Acquiring and Exploiting Lexical Knowledge for Twitter Sentiment Analysis

A thesis
submitted in partial fulfillment
of the requirements for the degree
of
Doctor of Philosophy

at
The University of Waikato

by
Felipe Bravo Márquez

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Department of Computer Science
Hamilton, New Zealand
July, 2017

Para Constanza y mis padres.

Abstract

The most popular sentiment analysis task in Twitter is the automatic classification of tweets into sentiment categories such as positive, negative, and neutral. State-of-the-art solutions to this problem are based on supervised machine learning models trained from manually annotated examples. These models are affected by label sparsity, because the manual annotation of tweets is labour-intensive and time-consuming.

This thesis addresses the label sparsity problem for Twitter polarity classification by automatically building two type of resources that can be exploited when labelled data is scarce: opinion lexicons, which are lists of words labelled by sentiment, and synthetically labelled tweets.

In the first part of the thesis, we induce Twitter-specific opinion lexicons by training words level classifiers using representations that exploit different sources of information: (a) the morphological information conveyed by part-of-speech (POS) tags, (b) associations between words and the sentiment expressed in the tweets that contain them, and (c) distributional representations calculated from unlabelled tweets. Experimental results show that the induced lexicons produce significant improvements over existing manually annotated lexicons for tweet-level polarity classification.

In the second part of the thesis, we develop distant supervision methods for generating synthetic training data for Twitter polarity classification by exploiting unlabelled tweets and prior lexical knowledge. Positive and negative training instances are generated by averaging unlabelled tweets annotated according to a given polarity lexicon. We study different mechanisms for selecting the candidate tweets to be averaged. Our experimental results show that the training data generated by the proposed models produce classifiers that perform significantly better than classifiers trained from tweets annotated with emoticons, a popular distant supervision approach for Twitter sentiment analysis.

Acknowledgments

I want to start this acknowledgment section with a story. In October 2012, I attended SPIRE, an information retrieval conference in Cartagena, Colombia, to present some work related to my master's thesis. One of the keynotes was Ian Witten from the University of Waikato, who I knew quite well from the WEKA workbench and the data mining book. We had a brief chat where he asked me about my future plans. I told him I was looking for a PhD program and he suggested the University of Waikato in New Zealand. Since that moment, the idea of moving to New Zealand was seeded into my mind.

Back in Chile, I contacted Eibe Frank from the machine learning lab at Waikato University via email. He asked me what sort of topic I wanted to investigate for a potential PhD thesis, to what I replied some vague ideas about doing sentiment analysis on Twitter streams.

Only a small fraction of the emails we receive are really important. From this small fraction there is an even smaller fraction capable of triggering significant changes in our lives. On the 18th of December of 2012 I received one of those life changing emails from Bernhard Pfahringer, from which I quote the following sentence: "I'd be very keen to supervise you on a PhD doing sentiment extraction on Twitter streams or similar".

It took a whole year of paperwork and application forms to make my trip to New Zealand possible. One day of that year, I visited the library of the University of Chile and took a copy of the data mining book authored by Witten and Frank. The last sentence of the acknowledgement section captured my attention. The sentence talked about how New Zealand had brought the authors and their corresponding families together and provided an ideal, even idyllic place to do that work.

Another life-changing email came up on the 17th of December of 2013. This time the sender was Gwenda from the Scholarships office of the University of Waikato. I was finally reading the sentence I was anxiously waiting to read: "We are delighted to inform you that your application for a Doctoral Scholarship has been successful." Constanza came to my place one hour later, we knew that our dream of living overseas was more real than ever.

Now, I find myself, some years later, writing the acknowledgements section of my PhD thesis. I am in the the same lab where the WEKA workbench and the data mining book were created. On my desk lies a copy of the recently printed fourth edition of the book. I am happily surprised to find my name mentioned in the acknowledgement section of this new edition. The final sentence of that section is still the same of the version I read years ago at the library in the University of Chile. This time it is me who can confirm with certainty that our machine learning lab in New Zealand was an idyllic place to work on the thesis you are reading right now.

Now is time for some acknowledgment.

I want to sincerely thank all the people who in one way or another contributed to the completion of this thesis.

First to Constanza, my partner who crossed the Pacific Ocean with me to make this dream possible. Living together in New Zealand has been the most rewarding experience of my life. I also want to express my gratitude to my parents who supported me from a distance in every aspect during all this period. This work is dedicated to you. My siblings, my extended family, and my close friends from Chile also deserve my gratitude.

I am thankful to my supervisors Bernhard Pfahringer and Eibe Frank who did an outstanding job supervising this thesis. They both were always available when I knocked on their doors, providing insightful guidance, allowing me to explore my own ideas, and furthermore their solid background in machine learning and programming was a permanent support for putting these ideas into practice. Thanks to Bernhard for giving me the chance of doing a PhD without even knowing me, and to Eibe for the thorough proofreading of all my drafts. I hope to follow your example if I have the chance to supervise someone someday.

I would like to thank all members of the Machine Learning group and all the visitors who made this a rewarding place to work, especially Brian Hardyment for being a true and supportive friend.

I want to express my gratitude to other good new friends I met in New Zealand outside academia: Juancho Martínez and Irina Goethe. Thanks a lot for the good moments we spent with Constanza with each of you and your corresponding families.

I want also thank to Saif Mohammad from NRC Canada. Thank you for sharing your NLP expertise and for being an excellent host when I visited you in Canada.

I would also like to thank the examiners of this thesis, Cécile Paris and Mike Thelwall, for taking the time of reading this thesis and for their valuable comments.

I want to thank all the anonymous reviewers of the papers submitted in this work, even those of the rejected ones. Many of the experiments you will find in this thesis were suggested by them.

I want to thank my former mentors and colleagues from Chile: Marcelo Mendoza, Bárbara Poblete, Claudio Gutiérrez, Mauricio Marín, Gastón L'huillier, Sebastián Ríos, and Juan Velásquez.

Even though he did not participate much in my PhD work, I am very thankful to Ian Witten, who initiated this aventure by suggesting the University of Waikato as a place for doing my PhD.

Last but not least, I would like to thank the financial support of the University of Waikato Doctoral Scholarship, and the machine learning group who funded all my conference trips.

Contents

1 Introduction 1
1.1 Twitter . . . 2
1.2 Classification . . . 3
1.2.1 Logistic Regression Models . . . 3
1.2.2 Support Vector Machines . . . 5
1.3 Research Problem . . . 7
1.4 Existing Solutions and their Limitations . . . 9
1.5 Research Proposal . . . 10
1.5.1 Word-sentiment Associations . . . 11
1.5.2 Tweet Centroid Model for Lexicon Induction . . . 12
1.5.3 Partitioned Tweet Centroids for Distant Superivison . . . 13
1.5.4 Annotate-Sample-Average . . . 13
1.6 Publications . . . 14
1.7 Experimental Methodology . . . 15
1.7.1 Evaluation Measures . . . 16
1.8 Thesis Outline . . . 19

2 Sentiment Analysis and Social Media 21
2.1 Primary Definitions . . . 21
2.2 Sentiment Classification of Documents, Sentences, and Tweets . . . 23
2.2.1 Supervised Approaches . . . 23
2.2.2 Lexicon-based Approaches . . . 26
2.2.3 Subjectivity Detection . . . 28
2.2.4 Multi-Domain Sentiment Classification . . . 28
2.2.5 Twitter Sentiment Analysis . . . 32
2.3 Polarity Lexicon Induction . . . 40
2.3.1 Semantic Networks . . . 40

2.3.2 Corpus-based approaches 42
2.4 Lexical Resources for Sentiment Analysis 45
2.4.1 Comparison of Lexicons 48
2.5 Analysis of Aggregated Social Media Opinions 52
2.5.1 Temporal Aspects of Opinions 52
2.5.2 Predictions using Social Media 54
2.6 Discussion 56

3 Word-sentiment Associations for Lexicon Induction 59
3.1 Proposed Method 61
3.1.1 Automatically-annotated Tweets 62
3.1.2 Word-level Features 66
3.1.3 Ground-Truth Word Polarities 69
3.2 Evaluation 70
3.2.1 Exploratory Analysis 70
3.2.2 Word-level Classification 73
3.2.3 Lexicon expansion 80
3.2.4 Extrinsic Evaluation of the Expanded Lexicons 81
3.3 Discussion 86

4 Distributional Models for Affective Lexicon Induction 89
4.1 Polarity Lexicon Induction with Tweet Centroids 90
4.1.1 Tweet Centroids and Word-Context Matrices 91
4.1.2 Evaluation 93
4.2 Inducing Word–Emotion Associations by Multi-label Classification 99
4.2.1 Multi-label Classification of Words into Emotions 101
4.2.2 Evaluation 102
4.3 Discussion 108

5 Transferring Sentiment Knowledge between Words and Tweets 111
5.1 Tweet-Centroids for Transfer learning 113
5.2 Experiments 116
5.2.1 The word-tweet sentiment interdependence relation 116
5.2.2 From opinion words to sentiment tweets 118

5.2.3 From tweets to opinion words 122
5.3 Discussion 125

6 Lexicon-based Distant Supervision: Annotate-Sample-Average 127
6.1 The Annotate-Sample-Average Algorithm 128
6.2 ASA and The Tweet Centroid Model 133
6.3 The Lexical Polarity Hypothesis 135
6.4 Classification Experiments 137
6.4.1 Sensitivity Analysis 141
6.4.2 Learning Curves 142
6.4.3 Qualitative Analysis 144
6.5 Discussion 145

7 Conclusions 147
7.1 Summary of Results 147
7.2 Contributions 149
7.3 Future Work 150
7.3.1 Extensions to the Word-Sentiment Association Method . . . 150
7.3.2 Extensions to the Tweet Centroid Model 150
7.3.3 Extensions to ASA 152

Bibliography 153

List of Figures

2.1 Venn diagrams of the overlap between opinion lexicons. a) lexicons created manually, and b) lexicons created automatically. 49

3.1 Twitter-lexicon induction process. The bird represents the Weka machine learning software. 62
3.2 Word-level time series. 71
3.3 PMI-SO vs SGD-SO scatterplot. 74
3.4 PMI-SO and SGD-SO Boxplots. 75
3.5 Word clouds of positive and negative words using log odds proportions. 81

4.1 Twitter-lexicon induction with tweet centroids. The bird represents the Weka machine learning software. 92
4.2 Emotion classification results obtained using word embeddings of different dimensionalities, generated from various window sizes. Maximum F1 is achieved for 400 by 5. 104
4.3 A visualisation for the expanded emotion lexicon. 106

5.1 Transfer Learning with tweet centroids. The bird represents the Weka machine learning software. 112
5.2 Violin plots of the polarity of tweets and words. 118
5.3 Word clouds of positive and negative words obtained from a message-level classifier. 124

6.1 Polarity distributions of *posT* and *negT*. 136
6.2 Heatmap of ASA parameters on the SemEval dataset. The highest F1 value for m=F is 0.76 ($a = 10$, instances = 200), and for m=T is 0.74 ($a = 5$, instances = 20). The highest AUC values for m=F and m=T occur with the same configurations as the highest values for F1 and are 0.85 and 0.81, respectively. 142
6.3 Learning curves for the SemEval dataset. 143

List of Tables

1.1 Classification confusion matrix. 17

2.1 Positive and negative emoticons. 33
2.2 Intersection of words. 49
2.3 Neutrality and uniqueness of each Lexicon. The lexicons are categorised according to the annotation mechanism. 50
2.4 Agreement of lexicons. 51
2.5 Sentiment values for different words. The scores in the SWN3 column correspond to the difference between the positive and negative probabilities assigned by SWN3 to the word. 52

3.1 Emoticon-annotated datasets. 64
3.2 Model transfer datasets with different threshold values. 66
3.3 Time series features. 68
3.4 Lexicon Statistics. 70
3.5 Word-level polarity classification datasets. 72
3.6 Word-level feature example. 73
3.7 Information gain values. Best result per column is given in bold. . . 76
3.8 World-level classification performance with emoticon-based annotation. Best result per row is given in bold. 77
3.9 Word classification performance using model transfer. Best result per column is given in bold. 78
3.10 World-level classification performance using model transfer. Best result per row is given in bold. 79
3.11 Example list of words in expanded lexicon. 80
3.12 Message-level polarity classification datasets. 82
3.13 Message-level polarity classification performance. Best result per column is given in bold. 84
3.14 Message-level polarity classification performance with outlier removal. Best result per columns is given in bold. 86

4.1 Dataset properties. 94
4.2 Word-level 2-class polarity classification performance. 94
4.3 Word-level three-class polarity classification performance. 95
4.4 Intrinsic Evaluation of tweet centroids and PPMI for lexicon induction. 96
4.5 Example of induced words. 97
4.6 Message-level classification performance. 98
4.7 Extrinsic Evaluation of TCM and PPMI for lexicon induction. 99
4.8 Word-level multi-label classification results. Best results per column for each performance measure are shown in bold. The symbol + corresponds to statistically significant improvements with respect to the baseline. 105
4.9 Message-level classification results over the Hashtag Emotion Corpus. Best results per column are given in bold. 107

5.1 Average number of positive and negative instances generated by different models from 10 collections of 2 million tweets. 121
5.2 Message-level Polarity Classification AUC values. Best results per column are given in bold. 121
5.3 Number of positive and negative words from AFINN. 123
5.4 Word-level polarity classification results for the AFINN lexicon. Best results per row are given in bold. 123

6.1 Probabilities of sampling a majority of tweets with the desired polarity. 132
6.2 Average number of positive and negative instances generated by different distant supervision models from 10 collections of 2 million tweets. 138
6.3 Macro-averaged F1 and AUC measures for different distant supervision models. Best results per column for each measure are given in bold. 139
6.4 Examples of tweets classified with ASA. Positive and negative words from AFINN are marked with blue and red colours respectively. The leftmost column indicates the classifier's output. 144

Chapter 1

Introduction

Social media platforms and, in particular, microblogging services[1] such as **Twitter**[2], **Tumblr**[3], and **Weibo**[4] are increasingly being adopted by users to access and publish information about a great variety of topics. These new mediums of expression enable people to connect to each other, and voice their opinion in a simple manner (Jansen, Zhang, Sobel and Chowdury, 2009).

Sentiment analysis or opinion mining refers to the application of techniques from fields such as natural language processing (NLP), information retrieval and machine learning, to identify and extract subjective information from textual datasets (Pang and Lee, 2008). One of the most popular sentiment analysis tasks is the automatic classification of documents or sentences into sentiment categories such as positive, negative, and neutral. These sentiment classes represent the writer's sentiment toward the topic addressed in the message.

Sentiment analysis applied to social media platforms has received increasing interest from the research community due to its importance in a wide range of fields such as business, sports, and politics. Several works claim that social phenomena such as stock prices, movie box-office revenues, and political elections, are reflected by social media data (Bollen, Mao and Zeng, 2011; Asur and Huberman, 2010; Gayo-Avello, 2013) and that opinions expressed in those platforms can be used to assess the public opinion indirectly (O'Connor, Balasubramanyan, Routledge and Smith, 2010).

This thesis focuses on analysing the sentiment of Twitter data. We use Twitter because it is the most widely-used microblogging service and provides large amounts of freely available public data. We propose machine learning-

[1] http://en.wikipedia.org/wiki/Microblogging
[2] http://www.twitter.com
[3] http://www.tumblr.com
[4] http://www.weibo.com

based models to tackle two sentiment analysis tasks: 1) classifying Twitter words into sentiment categories, and 2) training message-level polarity classifiers from unlabelled messages.

The remainder of this chapter is organised as follows. Section 1.1 presents a brief description of Twitter. Section 1.2 introduces the supervised machine learning methods used in this thesis. Section 1.3 states the research problem that it addresses. Existing solutions to the problem and their limitations are briefly presented in Section 1.4. The research proposal and the proposed methods are introduced in Section 1.5. The publications derived from this work are listed in Section 1.6. The experimental methodology used to evaluate the methods is presented in Section 1.7. In Section 1.8, we present an outline of the thesis' structure.

1.1 Twitter

Twitter is a microblogging service founded in 2006, in which users post messages or **tweets**. It was originally designed to be an SMS-based service where messages are restricted to 160 characters. Thus, tweets are limited to 140 characters, leaving 20 characters for the username. Twitter users may subscribe to the tweets posted by other users, an action referred to as "following". The service can be accessed through the Twitter website or through applications for smartphones and tablets.

Twitter users have adopted different conventions such as replies, retweets, and hashtags in their tweets. Twitter replies, denoted as `@username`, indicate that the tweet is a response to a tweet posted by another user. Retweets are used to re-publish the content of another tweet using the format `RT @username`. Hashtags are used to denote the context of the message by prefixing a word with a hash symbol e.g., #obama, #elections, etc. The size restriction and content sharing mechanisms of Twitter have created a unique dialect (Hu, Talamadupula and Kambhampati, 2013) that includes many abbreviations, acronyms, misspelled words, and emoticons that are not usual in traditional media, e.g., omg, loove, or :). Words and phrases that are frequently used during a particular time period are known as "trending topics". These topics are listed by the platform for different regions of the world, and can also be personalised to the user[5].

Twitter has become the most popular microblogging platform with hundreds

[5] https://support.twitter.com/articles/101125

of millions of users spreading millions of personal posts on a daily basis[6]. The rich and great volume of data propagated in it offers many opportunities for the study of public opinion and for analysing consumer trends (Jansen et al., 2009). Tweets published by public accounts can be freely retrieved using one of the two Twitter APIs: 1) the REST search API[7], which allows the submission of queries composed of key terms, and 2) the streaming API[8], from which a real-time sample of public posts can be retrieved. These APIs enable retrieval of domain-specific tweets restricted to certain words, users, geographical location, or time periods, in order to analyse tweets associated with a particular event or population sample.

1.2 Classification

Classification is the task of predicting a discrete variable y with L possible categories from examples represented by a set of features or independent variables $x_1, x_2, \ldots, x_n$ in a feature space $\mathcal{X}$. In order to train a classifier we need to learn a hypothesis function f from a collection of N training examples. This collection of records has the form (X, Y), and is usually referred to as the training dataset. Each entry of the dataset is a tuple (x, y), where x is the feature vector and y is the class or target label. When the possible outcomes of y are restricted to binary values, $y_i \in \{+1, -1\}$, $\forall i \in \{1, \ldots, N\}$, the classification problem is referred to as a binary classification problem.

The process of learning a hypothesis function from a training dataset is referred to as *supervised learning*, and there exist many machine learning algorithms for training such functions, many of which are described in (Witten, Frank and Hall, 2011). The methods used in this thesis are logistic regression models and support vector machines (SVMs), because they are known to perform well on text classification problems (Manning, Raghavan and Schütze, 2008).

1.2.1 Logistic Regression Models

Logistic regression models estimate the posterior probability $P(y|x)$ of a binary target variable y given the observed values of x by fitting a linear model to the data. The parameters of the model are formed by a vector of parameters w, which is related to the feature space $\mathcal{X}$ by a linear function. Assuming the

[6] https://about.twitter.com/company
[7] https://dev.twitter.com/rest/public/search
[8] https://dev.twitter.com/streaming/overview

intercept term is $x_0 = 1$, the linear function has the following form:

$$h_w(x) = \sum_{i=0}^{n} w_i x_i = w^T x \tag{1.1}$$

This function $h_w(x)$ is mapped into the interval $[0, 1]$ using the logistic or sigmoid function:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{1.2}$$

In ridge logistic regression, which is the method applied in this thesis, parameters w are determined by minimising the following L_2-regularised loss function from a given training dataset of N examples:

$$\min_w \quad \sum_{i=1}^{N} \log(1 + e^{-y_i w^T x_i}) + \frac{\lambda}{2} w^T w \tag{1.3}$$

The expression $\log(1 + e^{-y_i w^T x_i})$ corresponds to the log-likelihood of a probabilistic model in which y given x follows a Bernoulli distribution, and the parameter λ ($\lambda \geq 0$) is a user-specified regularisation parameter. Several algorithms can be used for optimising w given a training dataset. In this thesis we use the trust region Newton method (Lin, Weng and Keerthi, 2008) implemented in the *LibLinear* library (Fan, Chang, Hsieh, Wang and Lin, 2008). This implementation scales well to large datasets and high-dimensional feature-spaces.

Once the parameters are estimated, the posterior probability of a testing example is calculated according to the following expression:

$$P(y|x) = \frac{1}{1 + e^{-w^T x}} \tag{1.4}$$

The classification output is obtained by imposing a decision threshold on the posterior probability which is normally 0.5.

Logistic regression for multi-class classification, called multinomial logistic regression, uses the *softmax* function instead of the sigmoid function:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for} j \in \{1, \ldots, L\}. \tag{1.5}$$

A softmax regression model estimates one parameter vector w for each

class[9]. The posterior distribution of x for class j is calculated as follows:

$$P(y = j|x) = \frac{e^{w_j^T x}}{\sum_{l=1}^{L} e^{w_l^T x}} \tag{1.6}$$

The example is normally classified into the class with the highest probability.

Another approach for applying logistic regressions to multi-class problems, which is the one implemented in *LibLinear* and used in this thesis, is the one-vs-the-rest strategy proposed in (Crammer and Singer, 2002). In this strategy, a single classifier is trained per class, using the examples of that class as the positive instances and the remaining ones as the negative instances.

1.2.2 Support Vector Machines

A support vector machine (SVM) is a binary classifier aimed at finding a large-margin hyperplane ($\omega^T \cdot x + b$) that separates the two class values $y \in \{+1, -1\}$ according to the feature space represented by x. If the data is linearly separable, the optimal hyperplane is the one that maximises the margin between positive and negative observations in the training dataset formed by N observations. In the general case, the task of learning an SVM from a dataset is formalised as the following optimisation problem:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w + C \sum_{i}^{N} \xi_i \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \forall i \in \{1, \ldots, N\} \\ & \xi_i \geq 0 \; \forall i \in \{1, \ldots, N\} \end{aligned} \tag{1.7}$$

The objective function of the problem combines the length of the parameter vector and the errors $\sum_i^N \xi_i$. The parameter C is referred to as the "soft margin regularization parameter" and controls the sensitivity of the SVM to possible outliers.

It is also possible to make SVMs find non-linear patterns efficiently using the kernel trick. A function $\phi(x)$ that maps the feature space x into a high-dimensional space is used. This high-dimensional space is a Hilbert space, and the dot product $\phi(x) \cdot \phi(x')$ can be represented as a kernel function $K(x, x')$. A

[9] In actual implementations, the parameter vector for one of the classes can be dropped because it is redundant.

popular kernel function is the radial basis function kernel (RBF):

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right) \tag{1.8}$$

in which σ ($\sigma > 0$) is a free parameter that has to be tuned for each specific problem.

By using kernels, the hyperplane is calculated in the high-dimensional space ($\omega^T \cdot \phi(x) + b$). The dual formulation of the SVM allows replacing every dot product by a kernel function as is shown in the following expression:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \cdot K(x_i, x_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \forall i \in \{1, \ldots, N\} \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{1.9}$$

where the parameters $\alpha_i, i \in \{1, \ldots, N\}$ correspond to the Lagrange multipliers of the constrained optimisation problem. A popular algorithm for solving this quadratic programming problem is sequential minimal optimisation (Platt, 1998). In this thesis we use the implementation provided by the *LIBSVM* library (Chang and Lin, 2011). Once the parameters α have been determined, it is possible to classify a new observation x_j according to the following expression:

$$\text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i \cdot K(x_i, x_j) + b\right) \tag{1.10}$$

The calculation of the bias term b varies according to the the solver algorithm (Platt, 1998). A convenient property of SVMs is that the values of α will be different from zero only for a certain (usually small) number examples known as support vectors. Thus, SVMs only need to evaluate the kernel function between x_j and the support vectors.

SVMs can also be applied to multi-class problems, e.g., by using the one-vs-the-rest strategy introduced in the previous section.

1.3 Research Problem

We refer to message-level polarity classification as the task of automatically classifying tweets into sentiment categories. This problem has been successfully tackled by representing tweets from a corpus of hand-annotated examples using feature vectors and training classification algorithms on them (Mohammad, Kiritchenko and Zhu, 2013). A popular choice for building the feature space $\mathcal{X}$ is the vector space model (Salton, Wong and Yang, 1975), in which all the different words or unigrams found in the corpus are mapped into individual features. Word n-grams, which are consecutive sequences of n words, can also been used analogously. Each tweet is represented as a sparse vector whose active dimensions (dimensions that are different from zero) correspond to the words or n-grams found in the message. The values of each active dimension can be calculated using different weighting schemes, such as binary weights or frequency-based weights with different normalisation schemes.

The message-level sentiment label space $\mathcal{Y}$ corresponds to the different sentiment categories that can be expressed in a tweet, e.g., positive, negative, and neutral. Because sentiment is a subjective judgment, the ground-truth sentiment category of a tweet must be determined by a human evaluator, and hence, the manual annotation of tweets into sentiment classes is a time-consuming and labour-intensive task. We refer to this problem as the label sparsity problem. Because supervised machine learning models are impractical in the absence of labelled tweets, the label sparsity problem imposes practical limitations on using these techniques for classifying the sentiment of tweets.

Crowdsourcing tools such as Amazon Mechanical Turk[10] or CrowdFlower[11] allow clients to use human intelligence to perform tasks in exchange for a monetary payment set by the client. They have been successfully used for manually labelling tweets into sentiment classes (Nakov, Rosenthal, Kozareva, Stoyanov, Ritter and Wilson, 2013). Nevertheless, a classifier trained from a particular collection of manually annotated tweets will not necessarily perform well on tweets about topics that were not included in the training data or on tweets written in a different period of time. This is because the relation between messages and the corresponding sentiment label can change from one domain to another or over time. We refer to this problem as the sentiment

[10] https://www.mturk.com
[11] https://www.crowdflower.com

drift problem.

Social media opinions are expressed in different domains such as politics, products, movie reviews, sports, among others. More specifically, opinions are expressed about particular topics, entities or subjects of a certain domain. For example, “Barack Obama” is a specific entity of the domain “politics”.

The words and expressions that define the sentiment of a text passage are referred to in the literature as *opinion words* (Liu, 2012). For instance, *happy* is a positive opinion word and *sad* is a negative one. As has been studied in (Engström, 2004; Read, 2005) many opinion words are domain-dependent. That means that words or expressions that are considered as positive or negative for a certain domain will not necessarily have the same relevance or orientation in a different context. This situation is clarified in the following examples taken from real posts on Twitter:

1. For me the queue was pretty **small** and it was only a 20 minute wait I think but was so worth it!!! :D @raynwise

2. Odd spatiality in Stuttgart. Hotel room is so **small** I can barely turn around but surroundings are inhumanly vast & long under construction.

3. My girlfriend just called me to say good night because she **accident** (sic) fell asleep without saying it earlier :) #ShesTooCute

4. I got some RAGE over this #Harambe **accident**. This is why there should be NO zoos.

Here we can see that opinion words **small** and **accident** can be used to express opposite sentiment in different contexts. This is a manifestation of the sentiment drift problem, and its main consequence is that a sentiment classifier that was trained on data of a particular domain may not necessarily have the same classification performance for other topics or domains.

Temporal changes in the sentiment pattern are another manifestation of sentiment drift. The relation between messages and their corresponding sentiment label for a particular topic is non-stationary, i.e., it can change over time (Durant and Smith, 2007; Bifet and Frank, 2010; Bifet, Holmes and Pfahringer, 2011; Silva, Gomide, Veloso, Meira and Ferreira, 2011; Calais Guerra, Veloso, Meira Jr and Almeida, 2011; Guerra, Meira and Cardie, 2014). For instance, when an unexpected event associated with the topic occurs suddenly (e.g., a scandal linked to a public figure), new expressions conveying sentiment

can arise spontaneously, such as *#trumpwall* and *#PrayForParis*. Additionally, other existing words or expressions can change their frequency affecting the polarity pattern of the topic. Hence, the accuracy of a sentiment classifier affected by this change would decrease over time.

This problem was empirically studied in (Durant and Smith, 2007) by training sentiment classifiers using training and testing data from different time periods. The results indicated a significant decrease in the classification performance as the time difference between the training and the testing data was increased.

A possible approach to overcome the sentiment drift problem is to constantly update the sentiment classifier with new labelled data (Silva et al., 2011). However, as discussed in (Silva et al., 2011; Calais Guerra et al., 2011; Guerra et al., 2014), the high volume and sparsity of social streams make the continuous acquisition of sentiment labels, even using crowdsourcing tools, infeasible. The label sparsity and sentiment drift problems are connected.

The research problem considered in this thesis is how to derive accurate polarity classifiers for Twitter in label sparsity conditions without relying on the costly process of human annotation.

1.4 Existing Solutions and their Limitations

Opinion lexicons are a well known type of resource for supporting sentiment analysis of documents, especially when sentiment-annotated documents are scarce. An opinion lexicon is a list of terms or *opinion words* annotated according to sentiment categories such as positive and negative. Opinion lexicons can be used as prior lexical knowledge for calculating the sentiment of documents and tweets in an unsupervised fashion (Hatzivassiloglou and Wiebe, 2000; Taboada, Brooke, Tofiloski, Voll and Stede, 2011; Thelwall, Buckley and Paltoglou, 2012), and to extract low-dimensional message-level features, such as the number of words in the message matching each sentiment category, for training sentiment classifiers from small samples of annotated data (Bravo-Marquez, Mendoza and Poblete, 2014; Kouloumpis, Wilson and Moore, 2011; Mohammad et al., 2013; Jiang, Yu, Zhou, Liu and Zhao, 2011).

Opinion lexicons, however, suffer from similar shortcomings as supervised models for classifying the sentiment of tweets. The ground-truth sentiment of a word is a subjective judgment determined by a human, and the diversity of informal expressions found in Twitter makes the manually annotation of opinion words also an expensive and time-consuming task. Furthermore, opinion

lexicons are not immune to the sentiment drift phenomenon. Some word polarities can be inaccurate for certain domains, and they can also become obsolete due to temporal changes in the sentiment pattern.

An appealing strategy to address both the label sparsity and sentiment drift problems for message-level polarity classification in Twitter is distant supervision. Distant supervision models are heuristic labelling functions (Mintz, Bills, Snow and Jurafsky, 2009) used for automatically creating training data from unlabelled corpora. These models have been widely adopted for Twitter sentiment analysis because large amounts of unlabelled tweets can be easily obtained through the use of the Twitter API.

Theoretically speaking, distant supervision is a direct solution to the label sparsity problem as it replaces the human annotation labour. It can also potentially solve the sentiment drift problem because existing classifiers can be updated with more recently labelled examples or with tweets annotated from the domain in which a drift is being observed.

A well-known distant supervision approach for Twitter polarity classification is the emoticon-annotation approach, in which tweets with positive :) or negative :(emoticons are labelled according to the polarity indicated by the emoticon after removing the emoticon from the content (Read, 2005). This method is affected by the following limitations:

1. The removal of all tweets without emoticons may cause a loss of valuable information.

2. Emoticons are likely to produce misleading labels such as in the following example: *"you're not dating me? sad... tragic... for you at least :)"*,

3. There are many domains such as politics, in which emoticons are not frequently used to express positive and negative opinions, and hence, it is very difficult to obtain emoticon-annotated data from these domains.

As we can see, existing methods based on opinion lexicons and distant supervision exhibit major drawbacks when used for classifying the sentiment of tweets in label sparsity conditions.

1.5 Research Proposal

This thesis addresses the label sparsity problem for Twitter polarity classification by acquiring and exploiting lexical knowledge. The research hypothesis is as follows:

> "Polarity classification of tweets when training data is sparse can be successfully tackled through Twitter-specific polarity lexicons and lexicon-based distant supervision."

The problem of acquiring lexical knowledge in the form of opinion lexicons is referred to as polarity lexicon induction. We propose two Twitter-specific polarity lexicon induction methods based on word-level classification: 1) word-sentiment associations and 2) the tweet centroid model. We also propose two distant supervision methods that exploit existing opinion lexicons for building synthetically labelled data on which message-level polarity classifiers can be trained: 1) partitioned tweet centroids and 2) annotate-sample-average (ASA). We now briefly review these methods.

1.5.1 Word-sentiment Associations

This method combines information from automatically annotated tweets and existing hand-made opinion lexicons to induce a Twitter-specific opinion lexicon in a supervised fashion. The induced lexicon contains part-of-speech (POS) disambiguated entries (e.g., noun, verb, adjective) with a probability distribution for positive, negative, and neutral polarity classes.

To obtain this distribution using machine learning, word-level attributes are used based on (a) the morphological information conveyed by POS tags and (b) associations between words and the sentiment expressed in the tweets that contain them. The sentiment associations are modelled in two different ways: using point-wise-mutual-information semantic orientation (PMI-SO) (Turney, 2002), which is based on the point-wise mutual information between a word and tweet-level polarity classes, and using stochastic gradient descent semantic orientation (SGD-SO), which learns a linear relationship between words and sentiment.

The message-level sentiment labels are obtained automatically using emoticons and a transfer learning approach. The transfer learning approach enables learning of opinion words from tweets without emoticons by deploying a message-level classifier trained from tweets annotated with emoticons on a target collection of unlabelled tweets.

The training words are labelled by a seed lexicon formed by combining multiple hand-annotated lexicons, and the induced lexicon is obtained after deploying the trained word-level classifier on the remaining unlabelled words

from the corpus of tweets.

The experimental results show that the method outperforms the word-level polarity classification performance obtained by using PMI-SO alone. This is significant because PMI-SO is a state-of-the-art measure for establishing world-level sentiment.

1.5.2 Tweet Centroid Model for Lexicon Induction

The tweet centroid model is another word-level classification model for polarity lexicon induction, which in contrast to the previous method, does not necessarily depend on labelled tweets and can perform lexicon induction directly from a given corpus of unlabelled tweets.

The distributional hypothesis (Harris, 1954) states that words occurring in similar contexts have similar meanings. Distributional semantic models (Turney and Pantel, 2010) are used for representing lexical items such as words according to the context in which they occur. The tweet centroid model is a distributional representation that exploits the short nature of tweets by treating them as the whole contexts of words. In the tweet centroid model, tweets are represented by sparse vectors using standard natural language processing (NLP) features, such as unigrams and low-dimensional word-clusters, and words are represented as the centroids of the tweet vectors in which they appear.

The lexicon induction is conducted by training a word-level classifier using these centroids to form the instance space and a seed lexicon to label the training instances. The trained classifier is deployed on the remaining unlabelled words in the same way as in the previous model.

Experimental results show lexicons generated with the tweet centroid model produce valuable features for classifying the sentiment of tweets when compared with the original seed lexicon.

The model is also used to produce a more fine-grained word-level categorisation based on emotion categories, e.g., anger, fear, surprise, and joy. This is done by employing labels provided by an emotion-associated lexicon (Mohammad and Turney, 2013) and multi-label classification techniques.

The tweet centroid model allows message-level classifiers trained from sentiment-annotated tweets to be deployed on words for polarity lexicon induction because both tweets and words are represented by feature vectors of the same dimensionality and can also be labelled according to the same sentiment categories, e.g, positive and negative. This is useful in scenarios where no labelled

words are available for training a word-level classifier, but labelled tweets can be obtained instead.

1.5.3 Partitioned Tweet Centroids for Distant Superivison

Lexicon-based distant supervision methods automatically create message-level training data from unlabelled tweets by exploiting the prior sentiment knowledge provided by opinion lexicons. As lexicons are usually formed by more than a thousand words, lexicon-based methods can potentially exploit more data than a small number of positive and negative emoticons.

The tweet centroid model can also be used as a lexicon-based distant supervision method. As tweets and words are represented by the same feature vectors, a word-level classifier trained from a polarity lexicon and a corpus of unlabelled tweets can be used for classifying the sentiment of tweets represented by sparse feature vectors. In other words, the labelled word vectors correspond to lexicon-annotated training data for message-level polarity classification.

A drawback of this simple idea is that the number of labelled words for training the word-level classifier is limited to the number of words in the lexicon. In some scenarios, it is desirable to be able to create training datasets that increase in size when increasing the size of the source corpus of unlabelled tweets. This is because many classifiers perform better when trained from large datasets (Witten et al., 2011). We propose a simple modification to the tweet centroid model for increasing the number of labelled instances, yielding “partitioned tweet centroids". This modification is based on partitioning the tweets associated with each word into smaller disjoint subsets of a fixed size. The method calculates one centroid per partition, which is labelled according to the lexicon. The experimental results show that partitioned tweet centroids outperform the emoticon-annotation approach for message-level polarity classification.

1.5.4 Annotate-Sample-Average

Annotate-Sample-Average (ASA) is another lexicon-based distant supervision method for training polarity classifiers in Twitter in the absence of labelled data. ASA takes a collection of unlabelled tweets and a polarity lexicon composed of positive and negative words and creates synthetic labelled instances for message-level polarity classification. Each labelled instance is created by

sampling with replacement a number of tweets containing at least one word from the lexicon with the desired polarity, and averaging the feature vectors of the sampled tweets.

The rationale of the method is based on the hypothesis that a tweet containing an opinion word with a known polarity is more likely to express the same polarity than the opposite one. Consequently, averaging multiple tweets containing words with the same polarity increases the confidence of obtaining a vector located in the region of the target polarity.

This hypothesis is empirically validated, and the experimental results show that the training data generated by ASA (after tuning its parameters) produces a message-level classifier that performs significantly better than a classifier trained from tweets annotated with emoticons and a classifier trained, without any sampling and averaging, from tweets annotated according to the polarity of their words.

1.6 Publications

During the course of this project, the following peer-reviewed papers have been published in journals and conference proceedings:

1. F. Bravo-Marquez, E. Frank, and B. Pfahringer *Positive, Negative, or Neutral: Learning an Expanded Opinion Lexicon from Emoticon-annotated Tweets*, In *IJCAI '15: Proceedings of the 24th International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina 2015.

2. F. Bravo-Marquez, E. Frank, and B. Pfahringer *From Unlabelled Tweets to Twitter-specific Opinion Words*, In *SIGIR '15: Proceedings of the 38th International ACM SIGIR Conference on Research & Development in Information Retrieval*. Santiago, Chile 2015.

3. F. Bravo-Marquez, E. Frank, and B. Pfahringer *Building a Twitter Opinion Lexicon from Automatically-annotated Tweets*, In *Knowledge-Based Systems*. Volume 108, 15 September 2016, Pages 65 –– 78.

4. F. Bravo-Marquez, E. Frank, and B. Pfahringer *Annotate-Sample-Average (ASA): A New Distant Supervision Approach for Twitter Sentiment Analysis*, In *ECAI'16: Proceedings of the biennial European Conference on Artificial Intelligence*. The Hague, Netherlands 2016.

5. F. Bravo-Marquez, E. Frank, and B. Pfahringer *From opinion lexicons to sentiment classification of tweets and vice versa: a transfer learning approach*, In *WI'16: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Omaha, Nebraska, USA 2016.

6. F. Bravo-Marquez, E. Frank, S. Mohammad, and B. Pfahringer *Determining Word–Emotion Associations from Tweets by Multi-Label Classification*, In *WI'16: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Omaha, Nebraska, USA 2016.

1.7 Experimental Methodology

The methods proposed in this thesis are evaluated empirically on collections of manually-annotated data. We use hand-annotated lexicons as ground truth for the polarity lexicon induction methods and tweets that were manually annotated into sentiment classes for evaluating the distant supervision methods. All these datasets are described in Chapter 2.

When evaluating a machine learning classifier, it is not recommended to carry out the evaluation on the same data on which the classifier was trained. The performance obtained on the training data is likely to be biased and misleadingly optimistic, because some classifiers are prone to learn random noise. This phenomenon is known as over-fitting, and is commonly addressed by evaluating classifiers on independent testing examples that were not included for training.

In the "hold-out" approach, the training dataset is split into two independent training and testing datasets. The classifier is trained over the training set and then used to classify the values of the testing set. The predicted outputs are compared with their corresponding gold standard values.

A drawback of the "hold-out" approach is that all the examples within the testing set are not used for training purposes. As it has been discussed before, the labelled observations are often expensive to obtain, and hence it would be better to use all the available training examples. The k-fold cross-validation approach tackles this problem by randomly partitioning the training data into k folds of the same size, which are all stratified to maintain the same class distribution as the original dataset. Then, for each fold k, a classifier is trained over the remaining $k-1$ folds and evaluated over the retained one. The evaluation measures are averaged for all the folds ensuring that all observations are used for both training and evaluation purposes. Cross-validation gives a

more robust estimation of the classifier's performance on unseen data than the "hold-out" approach, and allows estimating the standard deviation of the performance across the folds. Additionally, the cross-validation procedure can be repeated multiple times using different random partitions of the folds (varying the random seed number), in order to get a better estimation of the classifier's performance.

Cross-validation can be used for statistically comparing the performance of two different classifiers on the same data. The average performance scores produced by two classifiers for all k folds, or $n \times k$ if the process is repeated n times, can be compared using statistical tests such as the paired t-student test. The null hypothesis is that there is no difference in the average performance of two classification schemes. In this thesis, we compare different word-level and message-level sentiment classification schemes with cross-validation using the corrected resampled paired t-student test with an α level of 0.05 (Nadeau and Bengio, 2003). This statistical test is a variant of the paired t-student test that corrects the problems that arise when different performance estimations are calculated from overlapping samples.

Cross-validation is not suitable for evaluating distant supervision schemes. This is because the automatically-labelled examples, such as tweets with emoticons, correspond to a biased sample of the real tweets (Go, Bhayani and Huang, 2009). We calculate the average performance of distant supervision classifiers deployed on an independent target collection of manually-annotated examples by varying the collection of unlabelled tweets from which the labelled training data is generated. The performance of two distant supervision methods estimated in this way is compared using the non-parametric paired Wilcoxon signed-rank test with the significance value set to 0.05.

1.7.1 Evaluation Measures

Next, we introduce the performance metrics used for evaluating classifiers. Considering a binary classifier f deployed on a testing dataset, four possible outcomes can be calculated: 1) correctly classified positive observations or True Positives (TP), 2) correctly classified negative observations or True Negatives (TN), 3) negative observations wrongly classified as positive or False Positives (FP), and 4) positive observations wrongly classified as negative or False Negatives (FN). These outputs are normally displayed in a confusion matrix C such as Table 1.1.

For multi-class problems, the confusion matrix is generalised to an $L \times L$ ma-

	$y = +1$	$y = -1$
$f(x) = +1$	TP	FP
$f(x) = -1$	FN	TN

Table 1.1: Classification confusion matrix.

trix C where L is the number of classes and N is the total number of examples in the dataset. A cell C_{ij} corresponds to the number of examples for which the classifier predicts class i and the real class is j. Note that all correctly classified examples lie on the diagonal of the matrix, and that the sum of all the elements of the matrix is equal to the total number of examples (N):

$$N = \sum_{i=1}^{L} \sum_{j=1}^{L} C_{ij} \tag{1.11}$$

The evaluation measures are described below:

- Accuracy, the overall percentage of correctly classified observations. For binary classification problems, it is calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1.12}$$

  and it is generalised to multi-class problems according to the following expression:

$$\text{Accuracy} = \frac{\sum_{i=1}^{L} C_{ii}}{N} \tag{1.13}$$

- Kappa statistic: In classification problems in which the class distribution is highly skewed towards a majority class, a classifier can yield a high accuracy by chance. The kappa statistic κ (Cohen, 1960) corrects this problem by normalising the classification accuracy according to the imbalance of the classes in the data. A classifier that is always correct will have a κ of one. Conversely, if it makes the right predictions with the same probability as a random classifier, the value of κ will be zero. It is calculated as follows:

$$\kappa = \frac{\text{Accuracy} - p_c}{1 - p_c} \tag{1.14}$$

  where p_c corresponds to the following expression:

$$p_c = \sum_{i=1}^{L} \left(\sum_{j=1}^{L} \frac{C_{ij}}{N} \cdot \sum_{j=1}^{L} \frac{C_{ji}}{N} \right) \tag{1.15}$$

- Precision, the fraction of correctly classified positive observations over all the observations classified as positive:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{1.16}$$

  In multi-class problems, multiple precisions are calculated using a one-vs-the-rest strategy and averaged. It is common to weight the scores according to the relative frequency of their corresponding classes. This approach applies for all the remaining measures in multi-class scenarios.

- Recall (also called sensitivity and true positive rate), the fraction of correctly classified positive observations over all the positive observations:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{1.17}$$

- F1-score, the harmonic mean between the precision and recall:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{1.18}$$

- Area Under the Curve (AUC): The receiver operating characteristic (ROC) curve or ROC curve plots the true positive rate (TPR), which is equivalent to recall, against the false positive rate (FPR), which is calculated as follows:

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{1.19}$$

  The multiple values of TPR and FPR required for building an ROC curve are obtained using different threshold settings for a same classifier. For example, in a logistic regression model, different decision thresholds for the posterior probability are considered. The area under the ROC curve (AUC) is a useful metric because it is independent of any specific value for the decision threshold. This area is 1 for a perfect classifier and 0.5 for a random one.

1.8 Thesis Outline

This thesis is structured as follows. A review of the literature on sentiment analysis and social media is presented in Chapter 2. The word-sentiment association method for polarity lexicon induction is described and evaluated in Chapter 3. Chapter 4 describes the tweet centroid model for polarity lexicon induction and for determining word-emotion associations. In Chapter 5, the tweet centroid model is used for transferring sentiment knowledge between words and tweets. The partitioned version of the model for distant supervision is also described in that chapter. The annotate-sample-average distant supervision method is described and evaluated in Chapter 6. Chapter 7 presents the main findings and contributions of this thesis, as well as a perspective for future work.

Chapter 2

Sentiment Analysis and Social Media

In the early stages of the Web, its content was usually published by website owners associated with traditional information sources such as news media and companies, among other organisations. Additionally, the content was mainly about “facts” which are objective statements on particular entities or topics. In the 2000s, the rise of Web 2.0 platforms (O’Reilly, 2007), e.g., blogs, online social networks and microblogging services, changed this situation by allowing users to generate and share textual content in a simpler way. This situation caused an explosive growth of subjective information (i.e., personal opinions) available on the Web, which in turn provided new opportunities for information system developers. As the factual information has been traditionally processed using techniques such as information retrieval and topic classification, different types of methods are required in order to process the “subjective" content generated by users. In this chapter, we give a review of those methods, which are commonly referred to in the research literature as *opinion mining* and *sentiment analysis* techniques. We discuss works addressing sentiment classification of documents, sentences, and tweets, as well as methods for polarity lexicon induction. Popular existing opinion lexicons are also reviewed and analysed. Moreover, we discuss work conducting aggregated analysis of opinions and applications of sentiment analysis and social media mining, including predictions about stock market prices and election outcomes. Finally, we provide a discussion of existing developments in the field in the context of the research problem addressed in this thesis.

2.1 Primary Definitions

Let d be an opinionated document (e.g., a product review) composed of a list of sentences $s_1, \ldots, s_n$. As stated in (Liu, 2009), the basic components of an opinion expressed in d are:

- *Entity*: can be a product, person, event, organisation, or topic on which an *opinion* is expressed (*opinion target*). An entity is composed of a hierarchy of components and sub-components where each *component* can have a set of *attributes*. For example, a cell phone is composed of a screen, a battery among other components, the attributes of which could be the size and the weight. For simplicity, components and attributes are both referred to as *aspects*.

- *Opinion holder:* the person or organisation that holds a specific opinion on a particular *entity*. While in reviews or blog posts the holders are usually the authors of the documents, in news articles the holders are commonly indicated explicitly (Bethard, Yu, Thornton, Hatzivassiloglou and Jurafsky, 2004).

- *Opinion*: a view, attitude, or appraisal of an *object* from an *opinion holder*. An opinion can have a positive, negative or neutral *orientation*, where the neutral orientation is commonly interpreted as no opinion. The orientation is also named *sentiment orientation*, *semantic orientation* (Turney, 2002), or *polarity*.

Considering the components of the opinions presented above, an opinion is defined as a quintuple $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ (Liu, 2010). Here, e_i is an entity, a_{ij} is an aspect of e_i and oo_{ijkl} is the opinion orientation of a_{ij} expressed by the holder h_k during time period t_l. Possible values for oo_{ijkl} are the categories positive, negative and neutral or different strength/intensity levels. In cases when the opinion refers to the whole entity, a_{ij} takes a special value named GENERAL.

It is important to consider that within an opinionated document, several opinions about different entities and also different holders can be found. In this context, a more general opinion mining problem can be addressed consisting of discovering all opinion quintuples $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ from a collection D of opinionated documents. These approaches are referred to as *aspect-based* or *feature-based* opinion mining methods (Liu, 2009). As we can see, working with opinionated documents involves tasks such as identifying entities, extracting aspects from the entities, the identification of opinion holders (Bethard et al., 2004), and the sentiment evaluation of the opinions (Pang and Lee, 2008).

In addition to the orientation or polarity, there are other affective dimensions by which opinions can be evaluated like *subjectivity* and *emotion*.

A sentence of a document is defined as subjective when it expresses personal feelings, views or beliefs. It is common to treat neutral sentences as objective and opinionated sentences as subjective.

Emotions are subjective feelings and thoughts. According to (Parrot, 2001) people have six primary emotions, which are: love, joy, surprise, anger, sadness, and fear. Another categorisation, proposed by Ekman (1992), is formed by 6 basic emotions: anger, fear, joy, sadness, surprise, and disgust, which was latter extended by Plutchik (2001) to include two additional emotion states: anticipation and trust.

The affective dimension of a document can be represented using different variable types. Nominal variables are used to represent hard associations with the affective dimension e.g., positive or non-positive, while ordinal and numeric variables are used to represent intensity or strength levels, such as weakly positive, strongly positive, or 40% negative.

2.2 Sentiment Classification of Documents, Sentences, and Tweets

The most popular task in sentiment analysis is document sentiment classification, which generalises the message-level polarity classification problem presented in Chapter 1 to documents of any length. A common assumption for simplifying the problem is that the target document expresses opinions about one single entity from one opinion holder (Liu, 2009). When sentiment classification is applied to a single sentence instead of to a whole document, the task is named *sentence-level sentiment classification* (Wilson, Wiebe and Hoffmann, 2005). In this section we review five approaches to sentiment classification: 1) supervised approaches, 2) lexicon-based approaches, 3) subjectivity detection, 4) multi-domain sentiment classification, and 5) sentiment classification of tweets.

2.2.1 Supervised Approaches

A popular approach is to model the problem as a supervised learning problem. The idea behind this approach is to train a function capable of determining the sentiment orientation of an unseen document using a corpus of documents labelled by sentiment. For example, the training and testing data can be obtained from websites of product reviews where each review is composed of a free text comment and a reviewer-assigned rating. A list of available training

corpora from opinion reviews can be found in (Pang and Lee, 2008). Afterwards, the text data and the ratings are transformed into a feature-vector and a target value respectively. For example, if the rating is on a 1-5 star scale, high-starred reviews can be labelled as positive opinions and low-starred reviews as negative in the same way. The problem can also be formulated as an ordinal regression problem using the number of stars as the target variable (Pang and Lee, 2005). In (Pang, Lee and Vaithyanathan, 2002), the authors trained a binary classifier (positive/negative) over movie reviews from the Internet Movie Database (IMDb). They used the following features: unigrams, bigrams, and part of speech tags, and the following learning algorithms: Support Vector Machines (SVM), Maximum Entropy Classifier, and Naive Bayes. The best average classification accuracy obtained through a three-fold cross-validation was 82.9%. This result was achieved using purely unigrams as features and an SVM as learning algorithm.

More recent models have adopted the "representation learning" approach of learning the document representation directly from the data using neural networks (Collobert and Weston, 2008). Word embeddings are a popular choice among those approaches. They are low-dimensional continuous dense word vectors trained from unlabelled document corpora capable of capturing rich semantic information. A state-of-the-art word embedding model is the skip-gram model (Mikolov, Sutskever, Chen, Corrado and Dean, 2013) implemented in the *Word2vec*[1] library. In this method, a neural network with one hidden layer is trained for predicting the words surrounding a centre word, within a window of size k that is shifted along the input corpus. The centre and surrounding k words correspond to the input and output layers of the network, respectively, and are represented by 1-hot vectors, which are vectors of the size of the vocabulary ($|V|$) with zero values in all entries except for the corresponding word index that receives a value of 1. Note that the output layer is formed by the concatenation of the k 1-hot vectors of the surrounding words. The hidden layer has a dimensionality l, which determines the size of the embeddings (normally $l \ll |V|$). The word-embedding for each word can be obtained in two ways: 1) from the projection matrix connecting the input layer with the hidden one, and 2) from the projection matrix connecting the hidden layer with the output one. The network is efficiently trained using an algorithm called "negative-sampling", and the rationale of the model is that words occurring in similar contexts will receive similar vectors. There is an-

[1] https://code.google.com/p/word2vec/

other word embedding model called continuous bag of words (CBOW), which is analogous to the skip-gram model after swapping its input and output layers. Thus, the learning task of CBOW consist of predicting a centre word given its surrounding words in a window.

A simple approach for using word-embeddings in sentence-level polarity classification is to use the average word vector as the feature representation of a sentence, which is latter used as input for training a sentence-level polarity classifier (Castellucci, Croce and Basili, 2015).

There are syntactic dependencies such as negations and but clauses known as "opinion shifters" (Liu, 2009) that can strongly alter the overall polarity of a sentence, e.g., "I didn't like the movie", "I like you but I'm married". Representations based on unigrams or averaging word embeddings lack information about the sentence structure. Hence, they are unable to capture long-distance dependencies in the passage. In the following, we discuss two representation learning techniques for modelling semantic compositionality in sentences that were successfully employed for sentiment analysis.

A recursive neural tensor network for learning the sentiment of pieces of texts of different granularities, such as words, phrases, and sentences, was proposed in (Socher, Perelygin, Wu, Chuang, Manning, Ng and Potts, 2013). The network was trained on a sentiment annotated treebank[2] of parsed sentences for learning compositional vectors of words and phrases. Every node in the parse tree receives a vector, and there is a matrix capturing how the meaning of adjacent nodes changes. The main drawback of this model is that it relies on parsing. Thus, it would be difficult to apply it to Twitter data because of the lack of Twitter-specific sentiment treebanks and robust constituency parsers for Twitter (Foster, Cetinoglu, Wagner, Le Roux, Nivre, Hogan and van Genabith, 2011).

A paragraph vector-embedding model that learns vectors for sequences of words of arbitrary length (e.g, sentences, paragraphs, or documents) without relying on parsing was proposed in (Le and Mikolov, 2014). The paragraph vectors are obtained by training a similar network as the one used for training the CBOW embeddings. The words surrounding a centre word in a window are used as input together with a paragraph-level vector for predict the centre word. The paragraph-vector acts as a memory token that is used for all the centre words in the paragraph during the training the phase.

The recursive neural tensor network and the paragraph-vector embedding

[2] http://nlp.stanford.edu/sentiment/treebank.html

were evaluated on the same movie review dataset used in (Pang et al., 2002), obtaining an accuracy of 85.4% and 87.8%, respectively. Both models outperformed the results obtained by classifiers trained on representations based on bag-of-words features.

Most sentiment analysis datasets are imbalanced in favour of positive examples (Li, Wang, Zhou and Lee, 2011). This is presumably because users are more likely to report positive than negative opinions. The shortcoming of training sentiment classifiers from imbalanced datasets is that many classification algorithms tend to predict test samples as the majority class (Japkowicz and Stephen, 2002) when trained from this type of data. A semi-supervised model for imbalanced sentiment classification is proposed in (Li et al., 2011). The model exploits both labelled and unlabelled documents by iteratively performing under-sampling of the majority class in a co-training framework using random subspaces of features.

2.2.2 Lexicon-based Approaches

In scenarios where training data is scarce, supervised models become impractical. Here we review models that exploit existing lexical knowledge about the sentiment of words for classifying the sentiment of documents.

We identify two types of lexicon-based approaches: 1) unsupervised models that do not require a training corpus of labelled documents, and 2) mixed models that combine lexical knowledge with labelled documents.

A common way of computing the polarity of a document relying only on lexicon knowledge is to aggregate the orientation values of the known opinion words found in the document (Hatzivassiloglou and Wiebe, 2000). A simple approach is to calculate the difference between positive and negative words, and label the document according to the difference's sign. More sophisticated aggregation functions including rules for negations and opinion shifters have also been proposed (Taboada et al., 2011; Thelwall et al., 2012).

A well-known model that uses the orientation of two opinion words and co-occurrence statistics obtained from a search engine is proposed in (Turney, 2002). First of all, a part-of-speech (POS) tagging is applied to all words of the document. POS tagging automatically identifies the linguistic category to which a word belongs within a sentence. Common POS categories are: noun, verb, adjective, adverb, pronoun, preposition, conjunction and interjection. The hypothesis of this work is that phrases containing a sequence of an adjective or an adverb as adjective followed by an adverb probably express an

opinion. Therefore, all sentences with a sequence of words that satisfy the pattern described above are extracted. The sentiment of each selected phrase is calculated using the point-wise mutual information (PMI) (Church and Hanks, 1990), which gives a measure of statistical independence between two words:

$$\mathrm{PMI}(term_1, term_2) = \log_2 \left(\frac{Pr(term_1 \wedge term_2)}{Pr(term_1) Pr(term_2)} \right). \tag{2.1}$$

In order to compute the semantic orientation, the PMI value of each phrase is calculated against a positive and a negative opinion word: “poor” and “excellent”. Then, as shown in Equation 2.2, the first value is subtracted from the second one to form the PMI-SO score.

$$\text{PMI-SO}(phrase) = \mathrm{PMI}(\text{phrase}, \text{“excellent”}) - \mathrm{PMI}(\text{phrase}, \text{“poor”}). \tag{2.2}$$

The probabilities of the PMI values are estimated using frequency counts, which are calculated as the number of hits returned by a search engine in response to a query composed of the sentence and the word “excellent” and another query using the word “poor” in the same way. Finally, the PMI-SO of a document is calculated as the average PMI-SO of the phrases within it. If this value is positive, the sentiment orientation of the document is labelled with the tag “positive”, otherwise it is labelled with a “negative” tag. The method exhibits high variability in performance when applied to different domains: it achieved an accuracy of 84% for auto-mobile reviews and 66% for movie reviews.

Next, we describe hybrid models that use both sentiment-annotated documents and opinion words for classifying the sentiment of documents.

The simplest approach is to use the opinion words for calculating aggregated features in supervised classification schemes, such as the number of positive and negative words found in a passage (Jiang et al., 2011; Kouloumpis et al., 2011; Zirn, Niepert, Stuckenschmidt and Strube, 2011). Lexicon-based features exhibit good generalisation properties as they include information about words that do not necessarily occur in the training data.

In (Sindhwani and Melville, 2008), words and documents are jointly represented by a bipartite graph of labelled and unlabelled nodes. The sentiment labels of words and documents are propagated to the unlabelled nodes using regularised least squares. In (Li, Zhang and Sindhwani, 2009), the term-document matrix associated with a corpus of documents is factorised into

three matrices specifying cluster labels for words and documents using a constrained non-negative tri-factorisation technique. Sentiment-annotated words and documents are introduced into the model as optimisation constraints. A generative naive Bayes model based on a polarity lexicon, which is then refined using sentiment-annotated documents, is proposed in (Melville, Gryc and Lawrence, 2009).

2.2.3 Subjectivity Detection

As has been seen above, sentiment classification usually assumes that documents are opinionated. However, in many cases a document within a collection contains only factual information, e.g., a news article. Furthermore, an opinionated document may contain several non-opinionated sentences. Hence, identifying the subjective sentences in a document is a relevant task that can be carried out before the sentiment classification.

The problem of determining whether a sentence is subjective or neutral is called *subjectivity classification* (Wiebe and Riloff, 2005). This problem can also be formulated as a supervised learning problem. In (Wiebe, Bruce and O'Hara, 1999) and (Yu and Hatzivassiloglou, 2003) a subjectivity classifier was trained using Naive Bayes where an accuracy of 97% was achieved on a corpus of journal articles. However, in the context of tweets, subjectivity detection has shown to be a harder problem than polarity classification (Bravo-Marquez et al., 2014).

Alternatively, the detection of subjectivity and polarity can be jointly addressed by treating the problem as a 3-class classification problem with classes: neutral, positive, and negative.

According to (Koppel and Schler, 2006), neutral data is crucial for training accurate polarity classifiers because, on the one hand, learning from only positive and negative documents will not generalise well to neutral examples, and on the other hand, neutral training data allows for a better classification of positive and negative documents.

2.2.4 Multi-Domain Sentiment Classification

The relationship between text and sentiment, as enunciated in Chapter 1, can vary from one domain to another. One of the first works studying this phenomenon was the thesis of Engström (2004). In that work, multiple sentiment classifiers were trained and tested on data from different domains (e.g., movie

reviews, automobiles). Results showed that classifiers trained on certain domains were unlikely to perform as well when tested on different domains.

A possible approach to tackle this problem is to include labelled data from all the domains to which the model will be applied in the training dataset. The problem of this approach is that the data distribution may vary from one domain to another, and hence, it is very difficult to design a robust multi-domain classifier (Glorot, Bordes and Bengio, 2011). Another simple solution is to train domain-specific classifiers for each domain. However, we know from the discussion of the label sparsity problem that obtaining labelled data from several domains is a costly process. Additionally, the sentiment patterns of different domains are not equally closely related to each other. For example, book and movie reviews are more related to each other than restaurant reviews. Hence, the naive approach of training individual classifiers for each domain may be inefficient because the knowledge provided by labelled instanced from similar domains is not exploited.

According to (Read, 2005), emoticons are, unlike opinion words, potentially domain-independent sentiment indicators. Thus, they could address the domain-dependency problem when used to label training data for supervised learning.

In (Wu and Huang, 2015), the multi-domain sentiment classification problem is addressed by jointly training a global sentiment model with multiple domain-specific ones, in which each model corresponds to a linear classifier. A convex loss function is optimised that considers different sources of information: 1) labelled documents from all target domains, 2) a global sentiment lexicon, 3) domain-specific sentiment lexicons for each domain, and 4) inter-domain similarities. The domain-specific lexicons are calculated from labelled examples of each domain using PMI semantic orientation and expanded to words occurring in domain-specific unlabelled data. The expansion is done by propagating the existing labels using a graph of associations between words. The inter-domain similarities are calculated in two ways: 1) using unigram-based textual similarities, and 2) by relying on the cosine similarity between domain-specific lexicons. The loss function is regularised using a combination of L_1 and L_2 norms and optimised using an accelerated algorithm. The novelty of this approach is that it explicitly exploits the fact that some domains share more sentiment information with others. The experimental results show that the proposed model outperforms the multi-domain sentiment classification accuracy of several existing multi-domain approaches.

A number of methods have been proposed to adapt sentiment classifiers from a source to a target domain. This strategy is suitable when the availability of labelled data is much higher in the source domain than in the target domain. In this direction, four model-transfer approaches were compared in (Aue and Gamon, 2005). The dataset consisted of a mixture of four different domains: movie reviews, book reviews, product support services web survey data, and knowledge base web survey data. The first three approaches apply SVMs classifiers with the following features: unigrams, bigrams, and trigrams. In the first approach, which is used as the baseline method, one single classifier is trained from all the domains. The second one follows a similar idea as the former, but the features are limited to the ones observed in the target domain. The third approach uses ensembles of classifiers from the domains with available labelled data. Finally, the fourth approach combines small amounts of labelled data with large amounts of unlabelled data in the target domain following an expectation maximisation (EM) learning strategy based on naive Bayes. Experiments were carried out using different numbers of labelled examples across the different domains, and results showed that the EM approach tends to achieve better accuracy than the others. The authors argue that this occurred because the EM method is the only one that takes advantage of unlabelled examples in the target domain.

Glorot et. al also exploited unlabelled data for domain adaptation in (Glorot et al., 2011), using a deep learning procedure. High-level representations are learnt in an unsupervised fashion from unlabelled data provided from multiple domains. This is carried out using Stacked Denoising Auto-Encoders with a sparse rectifier unit (Vincent, Larochelle, Bengio and Manzagol, 2008). Then, a linear SVM is trained on the transformed labelled data of the source domain and used to classify the testing data from the target domain. The hypothesis of the approach is that higher-level features are intermediate abstractions which are shared across different domains. Experimental results show that this approach can successfully perform domain adaptation on a dataset of 22 domains.

A different, but somewhat related problem is to simultaneously extract topics and opinions from a corpus of opinionated data on multiple topics. Probabilistic generative models were proposed in (Mei, Ling, Wondra, Su and Zhai, 2007) and (Lin and He, 2009) to model the generative process of words regarding both topics and sentiment polarities. These models extend the topic modelling approach (Blei, Ng and Jordan, 2003), in which it is assumed that

words in a corpus of documents are generated by a mixture of topics. The extension is based on the assumption that words within topics are also generated by a sentiment model that defines the polarity of the word.

The model proposed in (Mei et al., 2007), called the topic-sentiment mixture model (TSM), relies on a mixture of four multinomial distributions to describe the stochastic process in which words are generated from a corpus of opinionated documents about multiple topics. The distributions and the generative process are described as follows:

1. The background topic model θ_B captures common English words such as “the", “a", and “of".

2. The k topic models $\Theta = \{\theta_1, \ldots, \theta_k\}$ capture the neutral words related to the different topics in the collection.

3. The positive sentiment model θ_P models positive opinions.

4. The negative sentiment model θ_N captures negative opinions.

The generation of a document proceeds as follows in this model. First, it is randomly decided whether the current word is a common English word or not. If so, the word is drawn from θ_B. Otherwise, it is decided from which of the k topics the word will be sampled. Then, it is decided whether the word will describe the topic with a neutral, positive, or negative orientation. According to this decision, the word is drawn from either θ_i (i being the selected topic), θ_P, or θ_N, respectively. This process is repeated until all the words from the document are generated.

The parameters of the model are estimated using a maximum a posteriori estimation procedure. The prior distributions of the sentiment models are learnt first. Then, they are combined with the data likelihood to estimate the parameters of the maximum a posterior estimator. An important limitation of this model is that sentiment models are the same for all the different topics. Therefore, this model is not able to capture opinion words which are specific to particular domains.

Another sentiment topic model, called the joint topic sentiment model (JST) was proposed in (Lin and He, 2009). This model is unsupervised in the sense that is does not depend on documents labelled by sentiment. The words are drawn from a distribution jointly defined by the topics and the sentiment label. The model incorporates opinion words as prior information, and in contrast to TSM, it acknowledges that opinion words can be topic dependent.

2.2.5 Twitter Sentiment Analysis

Posts on Twitter (or tweets) are considered a rich resource for sentiment analysis. There is empirical evidence that Twitter users tend to post opinions about products or services (Pak and Paroubek, 2010), and tweets are short (at most 140-characters long) and usually straight to the point messages.

Opinion mining tasks that can be applied to Twitter data are polarity classification and opinion identification. Due to the short nature of tweets, a sentence-level classification approach can be adopted, assuming that tweets express opinions about one single entity. Furthermore, retrieving messages from Twitter is a straightforward task using the public Twitter API.

One of the main limitations of using supervised learning models for Twitter sentiment classification is the label sparsity problem introduced in Chapter 1. This problem was tackled in (Go et al., 2009) following the emoticon-annotation approach, in which emoticons are used as noisy indicator for labelling a training dataset in a distant supervision fashion. Smileys or emoticons are visual cues that are associated with emotional states (Carvalho, Sarmento, Silva and de Oliveira, 2009). The idea of using emoticons as labels was proposed in (Read, 2005) and is based on the idea that a textual passage containing a positive emoticon should have a positive orientation and the presence of a negative emoticon should indicate a negative orientation. In (Go et al., 2009), the Twitter API was used to retrieve tweets containing positive and negative emoticons, building a training dataset of $1,600,000$ tweets. Some emoticons which could be associated with positive and negative classes are presented in Table 2.1. The best accuracy obtained on a manually annotated dataset was of 83.7%. It was obtained using a maximum entropy classifier. The feature set was composed of unigrams and bigrams selected by the mutual information criterion. Furthermore, feature reduction was performed by replacing repeated letters (e.g., huuungry to huungry, loooove to loove) and replacing all mentions of Twitter users prefixed by the '@' symbol, with a generic token named "USER". In the same way URLs were replaced with a special token with the name "URL". Pak and Paraoubek conducted similar work in (Pak and Paroubek, 2010). They included, in addition to positive and negative classes obtained from emoticons, an objective or neutral class obtained from factual messages posted by Twitter accounts of popular newspapers and magazines. The corpus was used to train a 3-class classifier.

The noisy nature of emoticons when used as sentiment indicators for data labelling makes it very hard to achieve high performance with this approach

positive	negative
:)	:(
:-)	:-(
:D	=(
=)	:'(

Table 2.1: Positive and negative emoticons.

(Liu, Li and Guo, 2012). With the aim of taking advantage of both types of labels, emoticon-based and human-annotated, a language model that combines the two types was proposed in (Liu et al., 2012). Results showed that the integration of both resources produced better results than using them separately.

In (Zhang, Ghosh, Dekhil, Hsu and Liu, 2011), the authors proposed a lexicon-based approach for annotating unlabelled tweets with polarity classes regarding a given entity by aggregating the polarities of words from a lexicon with positive and negative words using a scoring function. The automatically labelled tweets are then used for training a classifier. This technique does not depend on supervision or manually labelled training data and is able to capture domain-specific sentiment patterns.

Another approach based on distant supervision and lexical prior knowledge is proposed in (Speriosu, Sudan, Upadhyay and Baldridge, 2011). The authors build a graph that has users, tweets, words, hashtags, and emoticons as its nodes. A subset of these nodes is labelled by prior sentiment knowledge provided by a polarity lexicon, the known polarity of emoticons, and a message-level classifier trained with emoticons. These sentiment labels are propagated throughout the graph using random walks.

A comprehensive survey of approaches exploiting unlabelled data for Twitter sentiment analysis based on self-training, co-training, topic modelling, and distant supervision is provided in (Silva, Coletta and Hruschka, 2016).

In (Kouloumpis et al., 2011) a supervised approach for Twitter sentiment classification is proposed based on linguistic features. In addition to using n-grams and part-of-speech tags as features, the authors use an opinion lexicon and particular characteristics from microblogging platforms such as the presence of emoticons, abbreviations and intensifiers. Empirical evaluations showed that although features created from the opinion lexicon are relevant, microblogging-oriented features are the most useful.

In 2013, The Semantic Evaluation (SemEval) workshop organised the "Senti-

ment Analysis in Twitter task"[3] (Nakov et al., 2013) with the aim of promoting research in social media sentiment analysis. The task was divided into two subtasks: the expression level and the message level. The former task is focused on determining the sentiment polarity of a message according to a marked entity within its content. In the latter task, the polarity has to be determined according to the overall message. The organisers released training and testing datasets for both tasks. The team that achieved the highest performance in both tasks among 44 teams was the *NRC-Canada* team (Mohammad et al., 2013). The team proposed a supervised method based on SVM classifiers using a hand-crafted features: word n-grams, character n-grams, part-of-speech tags, word clusters trained with the Brown clustering method (Brown, Desouza, Mercer, Pietra and Lai, 1992), the number of elongated words (words with one character repeated more than two times), the number of words with all characters in uppercase, presence of positive or negative emoticons, the number of individual negations, the number of contiguous sequences of dots, question marks and exclamation marks, and features derived from polarity lexicons (Mohammad et al., 2013). Two of these lexicons were generated automatically using large samples of tweets containing sentiment hashtags and emoticons (NRC-Hashtag and Sentiment140 lexicons). The mechanisms used for building those lexicons are detailed in Section 2.3.

In (Gonçalves, Araújo, Benevenuto and Cha, 2013) different sentiment analysis methods for polarity classification of social media messages are combined through an ensemble scheme. The authors weighted the methods according to their corresponding classification performance, showing that their combination achieves a better coverage of correctly classified messages. In a similar manner, existing lexical resources and methods for sentiment analysis were combined as meta-level features for supervised learning in (Bravo-Marquez, Mendoza and Poblete, 2013). The experimental results indicated that the combination of different resources provides significant improvement in accuracy for polarity and subjectivity classification.

Deep learning approaches have also been adopted for Twitter sentiment analysis. A supervised learning framework that uses sentiment-specific word embeddings and hand-crafted features was developed in (Tang, Wei, Qin, Liu and Zhou, 2014a). The word embeddings are obtained from emoticon-annotated tweets using a tailored neural network that captures the sentiment information of sentences and the syntactic contexts of words.

[3] http://www.cs.york.ac.uk/semeval-2013/task2/

A convolutional neural network architecture is developed in (Severyn and Moschitti, 2015b). Each tweet is represented as a matrix whose columns correspond to the words in the tweet, preserving the order in which they occur. The words are represented by dense vectors or embeddings trained from a large corpus of unlabelled tweets. The network is formed by the following layers: an input layer with the given tweet matrix, a single convolutional layer, a rectified linear activation function, a max pooling layer, and a softmax classification layer. The weights of the neural network are pre-trained using emoticon-annotated data, and then trained with the hand-annotated tweets from the SemEval competition. Experimental results show that the pre-training phase allows for a proper initialisation of the network's weights, and hence, has a positive impact on classification accuracy.

Incremental Approaches

The methods for Twitter sentiment analysis discussed so far do not consider the fact that the sentiment pattern can change over time due to sentiment-drift. A real-time sentiment classifier needs to be constantly updated in order to produce reliable results over time. Algorithms focused on learning from time evolving streams are referred to as "data stream mining models".

To the best of our knowledge, the first work studying social media opinions from a stream data mining point of view was (Bifet and Frank, 2010). Three fast incremental learning algorithms - Multinomial Naive Bayes, Stochastic Gradient Descent (SGD), and the Hoeffding Tree - were compared over two large collections of tweets: the emoticon-based training dataset used in (Go et al., 2009) and the Edinburgh corpus described in (Petrović, Osborne and Lavrenko, 2010). As the former dataset was formed purely of tweets with emoticons, in the second dataset only tweets with positive and negative emoticons were considered for training and testing the classifier. In this way, the training labels were acquired in both datasets, in the same way as in (Go et al., 2009). Furthermore, tweets were pre-processed using the same textual features as in previous batch-learning approaches (Go et al., 2009; Pak and Paroubek, 2010). The Massive Online Analysis (MOA) framework was used for the experiments using prequential accuracy and the kappa statistic for evaluation. The authors argued that the kappa measure is more suitable for unbalanced streams. The results indicated that, on the one hand, Hoeffding trees are not suitable for high-dimensional streams, and on the other hand, SGD and Naive Bayes perform comparably for this problem. A very strong

assumption made by this model is that sentiment labels are available across the entire stream. However, as a consequence of the label sparsity problem, it would be very difficult to continuously obtain labelled data to update the sentiment model and to properly address sentiment drift (Calais Guerra et al., 2011).

Bifet et al. developed MOA-TweetReader in (Bifet et al., 2011) as an extension of the MOA framework. This extension allows users to read tweets in real time, store the frequency of the most frequent terms, detect change in the frequency of words, and perform sentiment analysis in the same way as the aforementioned work.

A self-augmenting training procedure was proposed in (Silva et al., 2011). The learning task starts with a small sample of labelled examples used to train a classification rule learner. The classification model is formed by a set of rules of the form $\mathcal{X} \rightarrow s_i$, where the antecedent $\mathcal{X}$ corresponds to a set of terms and the consequent s_i is the predicted orientation. New messages are classified through a weighted vote based on the confidence values of all rules where the antecedent terms are observed. The confidence of a rule is the conditional probability of the polarity s_i given the terms in $\mathcal{X}$.

In this model, only the terms but not the polarity are known for future messages arriving from the stream. In order to deal with sentiment drift, the classification model is updated from unlabelled data in an incremental fashion. New rules are added to the classifier when the sentiment score calculated for arriving messages is higher than a user-specified threshold.

The model was tested on a dataset of hand-annotated tweets associated with different events that occurred in 2010. The results showed that the prediction performance remains stable, or even increases, as the data stream passes and new rules are extracted.

Another approach that also exploits unlabelled messages for updating a sentiment classifier was proposed in (Zimmermann, Ntoutsi and Spiliopoulou, 2014). The classifier adapts itself from unlabelled messages and additionally, introduces a mechanism to forget old messages. These tasks are referred to as *forward adaptation* and *backward adaptation* respectively.

The model takes as input a seed of labelled messages which are used to train a Multinomial Naives Bayes (MNB) classifier. In the forward adaptation step the arriving messages are classified with the initial classifier and evaluated according to a criterion of usefulness. This value corresponds to the difference in entropy of dataset before and after including the new labelled message. If

the usefulness value is above a threshold $\alpha \in (-1, 0)$ the parameters of the MNB classifier are updated. Afterwards, in the backward adaptation step, all word counts of the MNB classifier are updated using an exponential ageing function reducing the influence of old messages in the model.

We can see that both approaches, (Silva et al., 2011) and (Zimmermann et al., 2014), adapt the sentiment classifier from unlabelled data. The main difference is the way in which they consider the age of the messages. In (Silva et al., 2011) no distinction between old and new messages is made, and in (Zimmermann et al., 2014) the model discards old messages using an exponential ageing function.

There are two clear limitations for learning text-based models for time evolving sentiment analysis: sentiment drift and label sparsity. In order to tackle these limitations, a transfer learning approach was developed in (Calais Guerra et al., 2011). The idea behind transfer learning is to solve a source task which is somehow related to the target task, in scenarios where solving the former is much easier than solving the latter. The predictions made in the source domain are transferred to the target domain (Pan and Yang, 2010). In this proposal, a user-level analysis is transfered to solve a problem at the text-level. The target and the source tasks are the following: real time sentiment classification of social media messages, and social media user bias prediction, respectively.

The approach to predicting user bias is based on sociological theories claiming that humans tend to have biased opinions on different topics. The user bias towards a topic is quantified through social media endorsements. In the case of Twitter, user endorsements are represented by a directed graph, in which the vertices are users, and the edges (u, v) correspond to retweets made by user u of tweets posted by user v.

The problem is modelled as a relational learning task over a network of users connected by endorsements. The hypothesis is that similar users share a similar bias regarding a particular topic. Given a topic such as a political party or a sport team, a user can support K possibles sides in relation to it. Each user u in the network is represented by a bias vector $\overrightarrow{B_u} = [B_{u1}, \ldots, B_{uK}]$, where each element B_{ui} corresponds to the bias of the user towards the i-th side of the topic. In order to estimate the bias vector of all users in the network, a new graph called the Opinion Agreement Graph (OAG), is created. This is a weighted undirected graph where the vertices correspond to the users, and the weights of the edges u, v are calculated by averaging the following two

similarity measures:

1. The active similarity $\alpha(u, v)$, which measures the intensity of the endorsement of users u and v relative toward a given set of users. This measure is calculated using frequent pattern mining techniques, and is defined as the value of the *lift* measure of pair (u, v) in a database of transactions where each transaction contains the users who endorsed a given user. The lift measure compares the observed co-occurrence frequency of two elements with the expected co-occurrence frequency, assuming that both are mutually independent of each other.

2. The passive similarity $\rho(u, v)$, which measures the intensities of the endorsements of a given set of users towards users u and v. It is calculated in the same way as the active similarity but over a different database of transactions. The transactions of this database corresponds to all the users endorsed by a given user.

The bias vector of all nodes in the graph is calculated by propagating the vector of a few labelled users referred to as "attractor", who present a clear bias towards particular sides of the topic, e.g., official profiles of political candidates or political parties. The propagation is done by a random walk from the attractor nodes to the rest of the nodes in the OAG. This is based on the idea of propagating labels in graphs.

Once the user bias vectors of all users have been calculated, they are used for real-time polarity classification of arriving messages. The transfer is done through a propagation across terms assuming that words in a message will have a certain polarity toward an entity if they are adopted more frequently by users biased towards the same polarity. Assuming t is a term which is used to refer to a certain entity e, and let $U(t, e)$ be the set of users referring to the entity e using term t, a new vector $\overrightarrow{B}_{t,e}$ is calculated that consists of the sum vector of all users in $U(e, t)$:

$$\overrightarrow{B}_{t,e} = \sum_{u \in U(e,t)} \overrightarrow{B}_u$$

Let $\overrightarrow{B}^i_{t,e}$ be the strength of component i in vector $\overrightarrow{B}_{t,e}$. For each term in a message to be classified, the probability that the term refers to the entity e with a certain polarity i (e.g., positive or negative) is calculated according to

the following expression:

$$\hat{p}(\text{polarity} = i|t, e) = \frac{\overrightarrow{B}^{i}_{t,e}}{||\overrightarrow{B}_{t}||}$$

Finally, the overall polarity of the message is calculated by taking the polarity with the highest probability for the different terms in the message. This approach can deal with concept drift by incrementally updating $\overrightarrow{B}_{t,e}$ when new tweets arrive. The authors' argument is that the user bias vector is less susceptible to concept drift than a text-based sentiment pattern for a given topic.

The approach was tested over a dataset of tweets related to the Brazilian presidential election campaign of 2010 and the 12 most popular Brazilian soccer teams. The method was able to correctly classify 80% to 90% of the tweets by knowing the bias of 10% of the users who tweeted about the topics.

Guerra et al. (2014) proposed a method to obtain labelled sentiment data from a social media stream together with a feature representation suitable for dealing with sentiment drifts. Their approach follows the distant supervision paradigm, but instead of relying on emoticons or other text-based sentiment clues, it exploits social behaviour patterns usually observed in online social networks. These patterns are referred to as *self-report imbalances* and are defined as follows:

1. Positive-negative sentiment report imbalance: users tend to express positive feelings more frequently than negative ones.

2. Extreme-average sentiment report imbalance: users tend to express extreme feelings more than average feelings.

The first pattern is used to obtain labelled data for supervised classifiers on polarised topics, e.g., politics and sports. The idea is that a positive event for one group induces negative emotions toward the opposite group. Then, assuming that the size of the polarised groups for an event are known, the positive-negative imbalance is used to generate a probabilistic sentiment label for the event in a specific time frame. This is done by counting the number of users in the different groups that post a message during the time frame. It is important to note that these labels do not correspond to a particular message, but to the group of messages mentioning the entity in the time window. These labels represent uncertainty of the social context and can be used to predict the dominant sentiment during a time window.

The second pattern is used to create a feature representation named *text arousal* focused on terms appearing at spikes in the social stream. Time windows have a varying volume of messages, and further, according to the extreme-average report imbalance, spikes of activity in the stream tend to contain highly emotional words. Therefore, emotional words, which in turn are the most informative words for sentiment classification, will be more likely to occur during spikes of activity in the social network. For this reason, the feature representation includes the number of times that a term appears in high-volume time windows.

The *text arousal* model was compared with static representations based on term-frequency, and results indicated that it is more suitable to capture sentiment drifts. Finally, the overall method based on self-report imbalances was tested on sport events in Twitter, achieving accuracies of up to 84%.

2.3 Polarity Lexicon Induction

We studied in Section 2.2.2 how lexical knowledge about the sentiment of words can benefit the polarity classification of documents, sentences, and tweets. An opinion, polarity, or sentiment lexicon is a dictionary of opinion words with their corresponding sentiment categories or semantic orientations. A semantic orientation is a numerical measure for representing the polarity and strength of words or expressions. Lexicons can be manually created by taking words from different sources, and determining their sentiment values by human judgements. As this task can be very labour intensive, the labelling can be conducted through crowdsourcing mechanisms. An alternative approach is to build the lexicon automatically, which can be done by exploiting two types of resources: semantic networks and document collections. Previous work on opinion lexicon induction from these two type of resources is presented in the following two subsections. Afterwards, we describe popular existing lexicons for sentiment analysis and analyse how these resources interact with each other.

2.3.1 Semantic Networks

A semantic network is a network that represents semantic relations between concepts. The simplest lexicon induction approach, based on a semantic network of words such as WordNet[4], is to expand a seed lexicon of labelled opin-

[4] http://wordnet.princeton.edu/

ion words using synonyms and antonyms from the lexical relations (Hu and Liu, 2004; Kim and Hovy, 2004). The hypothesis behind this approach is that synonyms have the same polarity and antonyms have the opposite one. This process is normally iterated several times. In (Kamps, Marx, Mokken and De Rijke, 2004), a graph is created using WordNet adjectives as vertices and the synonym relations as edges. The orientation of a term is determined by its relative distance from the two seed terms *good* and *bad*. In (Esuli and Sebastiani, 2005), a supervised classifier is trained using a seed of labelled words that is obtained through expansion based on synonyms and antonyms. For each word, a vector space model is created from the definition or *gloss* provided by the WordNet dictionary. This representation is used to train a word-level classifier that is used for lexicon induction. An equivalent approach was applied later to create SentiWordNet[5] (Baccianella, Esuli and Sebastiani, 2010; Esuli and Sebastiani, 2006). In SentiWordNet, each WordNet *synset* or group of synonyms is assigned to classes *positive*, *negative* and *neutral*, with soft labels in the range $[0, 1]$.

Another well-known lexical resource for sentiment analysis built from concept-level semantic networks is SenticNet[6], which labels multi-word concepts according to both affective and semantic information. SenticNet is based on the *sentic computing* paradigm, which focuses on a semantics-preserving representation of natural language concepts and sentence structure (Cambria and Hussain, 2015). Multiple techniques have been exploited along the different versions of SenticNet. The first two versions were built using graph-mining and dimensionality-reduction techniques, and the third version integrates multiple knowledge sources by setting up pathways between concepts.

The automatic processing of emotions is the main focus of the field of *Affective Computing*, which is closely related to sentiment analysis (Cambria, 2016). WordNet-Affect[7] is a semantic network for affective computing in which some WordNet synsets are mapped into affective states corresponding to emotion and mood (Valitutti, 2004). WordNet-Affect was used together with SenticNet for building *EmoSenticSpace* (Poria, Gelbukh, Cambria, Hussain and Huang, 2014), a knowledge-base of natural language concepts annotated with emotion labels and polarity scores. This resource was built using fuzzy c-means clustering and support vector machine (SVM) classification.

[5] http://sentiwordnet.isti.cnr.it/
[6] http://sentic.net/
[7] http://wndomains.fbk.eu/wnaffect.html

ConceptNet[8] is a semantic network of commonsense knowledge formed by over 1.6 million assertions composed of two concepts connected by a relation e.g., *car usedFor driving*. There are 33 different types of relations such as IsA, PartOf and UsedFor. Lexicon expansion methods based on this resource were proposed in (Tsai, Wu, Tsai and Hsu, 2013; Weichselbraun, Gindl and Scharl, 2014; Wu and Tsai, 2014). In (Tsai et al., 2013), each concept on ConceptNet is given a sentiment score using iterative regressions that are then propagated via random walks. However, considering that not all relations from ConceptNet are necessarily related to sentiment, the model was further improved in (Wu and Tsai, 2014) using sequential forward search to find the best combination of sentiment-associated relations from ConceptNet. The model performs a bias correction step after the random walk process to reduce the variability in the obtained polarities.

A drawback of opinion lexicons is their lack of contextual information. A method for contextualising and interpreting ambiguous sentiment terms in opinion lexicons is proposed in (Weichselbraun et al., 2014). The method performs three steps to add positive and negative context terms to extend the expressiveness of the target resource: 1) identify ambiguous sentiment terms from SenticNet, 2) extract context information from a domain-specific corpus, and 3) associate the extracted context information with knowledge sources such as ConceptNet and WordNet.

Lexicons built from semantic networks are unable to capture sentiment information from words or concepts that go beyond the exploited network. Because words or concepts included in semantic networks such as WordNet and ConceptNet are based on formal English rather than informal expressions, the resources expanded from these networks will exhibit limitations when used with Twitter.

2.3.2 Corpus-based approaches

Corpus-based approaches exploit syntactic or co-occurrence patterns to induce a lexicon based on the words found within a collection of unstructured text documents.

In (Hatzivassiloglou and McKeown, 1997) the authors started with a set of adjectives whose semantic orientation was known a priori and then discovered new adjectives with their semantic orientations from a corpus by applying some linguistic conventions. They show, using a log-linear regression, that

[8] http://conceptnet5.media.mit.edu/

conjunctions between adjectives provide indirect information about the orientation. For example, adjectives connected with the conjunction “and” tend to have the same orientation and adjectives connected with the conjunction “but” tend to have the opposite orientation. This approach enables the extraction of domain-dependent information and the adaptation to new domains when the corpus of documents is changed.

The PMI-SO score, which was previously introduced in Section 2.2.2 (Equation 2.2) for estimating the semantic orientation of phrases, has been widely used for lexicon induction. Recapitulating, the PMI-SO of a target word or phrase corresponds to the difference between two PMI scores: 1) the PMI of the target word with a positive sentiment, and 2) the PMI of the target word with a negative sentiment. Apart from the original idea of counting the number of hits returned by a search engine in response to the target word together with known positive and negative words (Turney, 2002), other similar approaches have been used for Twitter lexicon induction based on associations between words and message-level sentiment labels (Becker, Erhart, Skiba and Matula, 2013; Kiritchenko, Zhu and Mohammad, 2014; Mohammad et al., 2013; Zhou, Zhang and Sanderson, 2014). In (Becker et al., 2013), tweets are labelled with a classifier trained from manually-annotated tweets using thresholds for the different classes to ensure high precision. In (Zhou et al., 2014), the emoticon-annotation approach is used to create domain-specific lexicons. In (Kiritchenko et al., 2014; Mohammad et al., 2013), tweets are labelled with emoticons and hashtags associated with emotions to create two different lexicons. These lexicons are tested for tweet-level polarity classification.

In (Mohammad and Kiritchenko, 2015), the authors collected around $50,000$ tweets with hashtags corresponding to the six Elkman emotions: #anger, #disgust, #fear, #happy, #sadness, and #surprise, referred to as the Hashtag Emotion Corpus. This corpus was used for creating a Twitter-specific emotion-association lexicon by mapping all the unigrams and bigrams from the corpus into strength association scores related to the six Elkman emotions. The scores between a word w and an emotion e are calculated based on PMI:

$$\mathrm{SoA}(w, e) = \mathrm{PMI}(w, e) - \mathrm{PMI}(w, \neg e)$$

The same article describes the creation of a fine-grained Twitter-oriented emotion lexicon created in an analogous way for 585 different emotion-associated hashtags.

Bahrainian, Liwicki and Dengel (2014) proposed another corpus-based model

based on an unsupervised message-level sentiment classifier and the Twitter API. The message-level classifier is based on a seed lexicon and opinion rules for handling intensifiers, diminishers, and negations. For each target word to be included in the lexicon, a set of tweets containing the word is retrieved by sending it to the API. Then, the word is classified by averaging the predicted sentiment obtained by the message-level classifier for the retrieved tweets.

In (Severyn and Moschitti, 2015a) words are used as features for predicting the polarity of emoticon-annotated tweets using a linear SVM. The SVM weight's are interpreted as word-level sentiment associations in the same way as in (Bifet and Frank, 2010). A similar approach is followed in (Vo and Zhang, 2016) using neural networks, in which each word receives a positive and negative weight.

An alternative approach is to represent Twitter words as embedding vectors that are classified into sentiment classes using machine learning and a seed lexicon to label the training data. In (Amir, Ling, Astudillo, Martins, Silva and Trancoso, 2015), state-of-the-art word embeddings such as skip-grams (Mikolov et al., 2013), continuous bag-of-words (Mikolov et al., 2013), and Glove (Pennington, Socher and Manning, 2014) were used as features in a regression model to determine the association between Twitter words and positive sentiment. In (Tang, Wei, Qin, Zhou and Liu, 2014b), a hybrid loss function for learning sentiment-specific word embeddings is proposed. The embeddings are obtained by combining syntactic information provided by the skip-gram model (Mikolov et al., 2013) and sentiment information provided by emoticon-annotated tweets. A transfer learning approach is followed in (Castellucci et al., 2015), by transferring sentiment labels from tweets to words. Words are represented by skip-gram embeddings whereas tweets are represented as the sum of the word vectors appearing in it. Note that in this way words and tweets reside in the same space. A message-level polarity classifier is trained from emoticon-annotated tweets and deployed on the word vectors to perform the lexicon induction.

A model for creating domain-specific opinion lexicons is proposed in (Hamilton, Clark, Leskovec and Jurafsky, 2016). Words from a source corpus of unlabelled text data (not necessarily tweets) are represented by embedding vectors. A square matrix of size $|V| \times |V|$ (V is the vocabulary), is built where each entry (i, j) corresponds to a smoothed PMI score of the word pair w_i,w_j based on co-occurrence counts within fixed-size sliding windows of text. This matrix is projected onto a low-dimensional space using singular value decomposition

(SVD) and used for building a graph of words associations. Each word in the graph is connected with its k most similar words according to cosine similarity in the low-dimensional space. The sentiment induction is carried out by propagating the known polarities of a seed lexicon to the remaining nodes in the graph using random walks. The model is used for creating domain-specific sentiment lexicons for different *Reddit*[9] communities. Examples of words exhibiting different polarities in different communities are the words "soft" and "animal", which are positive in a community dedicated to female perspectives and gender issues but negative in sports. Conversely, the words "crazy" and "insane" exhibit contradictory polarities in both domains. The same approach was also used for studying the evolution of opinion words by building lexicons from documents from the Corpus of Historical American English[10] written in consecutive decades between 1850 to 2000. The authors found that several words have changed their polarity over time, for instance the word "terrific" has changed from a negative to a positive polarity in the last decades. These results provide further evidence supporting the sentiment-drift problem discussed in Chapter 1.

2.4 Lexical Resources for Sentiment Analysis

In the following, we describe some of the most popular lexical resources for sentiment analysis.

OpinionFinder or MPQA Subjectivity Lexicon This is a hand-made lexical resource created by Wilson et al. (2005). It is part of OpinionFinder[11], a system that automatically detects subjective sentences in document corpora. A group of human annotators tagged words and phrases from a corpus of documents according to the polarity classes positive, negative, and neutral. A pruning phase was conducted over the dataset to eliminate tags with low agreement. Thus, a list of sentences and single words with their polarity tags was consolidated. The single words (unigrams) tagged as positive or negative correspond to a list of 6,884 English words. The lexicon also includes 17 words with mixed positive and negative polarities tagged as "both".

[9] https://www.reddit.com/
[10] http://corpus.byu.edu/coha/
[11] http://mpqa.cs.pitt.edu/opinionfinder/opinionfinder_2/

Bing Liu's Opinion Lexicon This lexicon is maintained and distributed by Bing Liu[12] and was used in several papers authored or co-authored by him (Liu, 2012). The lexicon consists of 2,006 positive words and 4,683 negative words. It includes misspelled words, slang words and some morphological variants.

ANEW Lexicon The *Affective Norms for English Words* lexicon (ANEW) proposed by Bradley and Lang (Bradley and Lang, 1999) provides emotional ratings for around 1,000 English words. These ratings are calculated according to three different psychological reactions of a person to a specific word: valence (the level of pleasantness), dominance (the degree of control), and arousal (the intensity of emotion). The reaction "valence", which ranges in the scale from pleasant to unpleasant, is the most useful value for polarity calculation.

AFINN Lexicon Inspired by ANEW, Nielsen (Årup Nielsen, 2011) created the *AFINN* lexicon, which is more focused on the language used in microblogging platforms. ANEW was released before the rise of microblogging and hence, many slang words commonly used in social media were not included. Considering that there is empirical evidence about significant differences between microblogging words and the language used in other domains (Baeza-Yates and Rello, 2011), a new version of ANEW was required. The word list includes slang and obscene words and also acronyms and Web jargon. Positive words are scored from 1 to 5 and negative words from -1 to -5. The lexicon includes 2,477 English words.

SentiWordNet Lexicon Already discussed in Section 2.3, SentiWordNet 3.0 (*SWN3*) is an improvement over the original SentiWordNet proposed in (Esuli and Sebastiani, 2006). It is based on *WordNet*, the well-known lexical database for English where words are clustered into groups of synonyms known as *synsets* (Miller, Beckwith, Fellbaum, Gross and Miller, 1990). In SentiWordNet each synset is automatically annotated in the range $[0, 1]$ according to positivity, negativity and neutrality.

Harvard General Inquirer The Harvard General Inquirer is a lexicon developed by Stone, Dunphy, Smith and Ogilvie (1966). The words of the lexicon are tagged according to multiple dimensions such as polarity, emotions, and

[12] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

semantics. The lexicon has 1,915 and 2,291 positive and negative words respectively.

NRC word-emotion association Lexicon This lexicon contains more than 14,000 distinct English words annotated according to both emotion and sentiment categories using the crowdsourcing Amazon Mechanical Turk platform. The emotion categories come from the Plutchick wheel of emotions (Plutchik, 2001) with categories joy, trust, sadness, anger, surprise, fear, anticipation, and disgust. The sentiment categories correspond to positive and negative polarities. All these categories are not mutually exclusive, and hence, a word can be tagged according to multiple emotions or polarities. Additionally, there are neutral words that are not associated with any emotion or polarity category.

NRC-Hashtag The NRC-Hashtag Sentiment Lexicon is an automatically created sentiment lexicon which was built from a collection of 775,310 tweets that contain positive or negative hashtags such as #good, #excellent, #bad, and #terrible. The tweets are labelled as positive or negative according to the hashtag's polarities. A sentiment score is calculated for all the words and bigrams found in the collection using the point wise mutual information (PMI) measure between each word and the corresponding polarity label of the tweet. The resource was created by the *NRC-Canada* team that won the SemEval task for Twitter polarity classification (Mohammad et al., 2013).

Sentiment140 Lexicon This lexicon was also provided by the *NRC-Canada* team and was created following the same approach used for creating the *NRC-Hashtag* lexicon. Instead of using hashtags as tweet labels, a corpus of 1.6 million tweets with positive and negative emoticons was used to calculate the sentiment words. The tweet collection is the same one as the one used for training the classifier proposed in (Go et al., 2009).

SentiStrength This is a lexicon-based sentiment analysis method[13] that returns positive and negative numerical scores for a given text passage (Thelwall et al., 2012). The positive score ranges from 1 (not positive) to 5 (extremely positive) and the negative one ranges from -1 (not negative) to -5 (extremely negative). The lexicon is hand-annotated and includes both formal English words and informal words used in social media (e.g., luv and lol), scored by

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

sentiment. The scores can also be adapted to a specific domain using machine learning. SentiStrength applies linguistic rules for dealing with negations, questions, booster words, and emoticons. These are used together with the lexicon for computing the positive and negative outputs.

SenticNet This is a concept-based semantic network for sentiment analysis, which we have already discussed in Section 2.3. SenticNet provides both sentiment and semantic information from over $30,000$ common sense knowledge concepts. It also provides a parser that returns the following two sentiment variables associated with each of the concepts found in a given message: the polarity score, and the sentic vector. The polarity score is a real value. The sentic vector is composed of emotion-oriented scores regarding the following emotions: pleasantness, attention, sensitivity, and aptitude. These dimensions are based on the Hourglass model of emotions (Cambria, Livingstone and Hussain, 2012), which in turn, is inspired by Plutchik's studies on human emotions.

2.4.1 Comparison of Lexicons

In this section we compare seven popular lexical resources for sentiment analysis: SWN3, NRC-emotion, OpinionFinder, AFINN, Liu Lexicon, NRC-Hashtag, and S140Lex. The aim of this study is to understand which type of information is provided by these resources and how they are related to each other. The lexicons may be compared according to different criteria: their sentiment scope or the type of variable used for representing their affective values, the approach used to build them, and the words that they contain. Regarding the sentiment scope, we have two lexicons with nominal polarity categories: OpinionFinder and Liu, four lexicons with numerical polarity scores indicating sentiment strength: AFINN, SWN3, NRC-hash, and S140Lex, and one multi-labelled lexicon with binary emotion-oriented associations: NRC-emotion, which also provides nominal polarity values.

Regarding the mechanisms used to build the lexicons, there are manually and automatically created resources. The lexicons Liu, AFINN, OpinionFinder, and NRC-emotion were manually created. They were created by taking words from different sources, and their sentiment values were mostly determined by human judgements using tools such as crowdsourcing in the case of NRC-emotion. SWN3 is a resource created automatically, whose words were taken from a semantic network (WordNet synsets) and its sentiment values computed using machine learning (Section 2.3). The lexicons NRC-hash and S140Lex

were automatically built from tweets annotated with hashtags and emoticons, respectively.

Intersection	OpFinder	AFINN	S140Lex	NRC-hash	Liu	SWN3	NRC-emotion
OpFinder	6,884	×	×	×	×	×	×
AFINN	1,245	2,484	×	×	×	×	×
S140Lex	3,460	1,789	60,113	×	×	×	×
NRC-hash	3,541	1,816	27,012	42,586	×	×	×
Liu	5,413	1,313	3,268	3,312	6,783	×	×
SWN3	6,199	1,783	16,845	17,314	5,480	146,977	×
NRC-emotion	3,596	1,207	8,815	8,995	3,024	13,634	14,182

Table 2.2: Intersection of words.

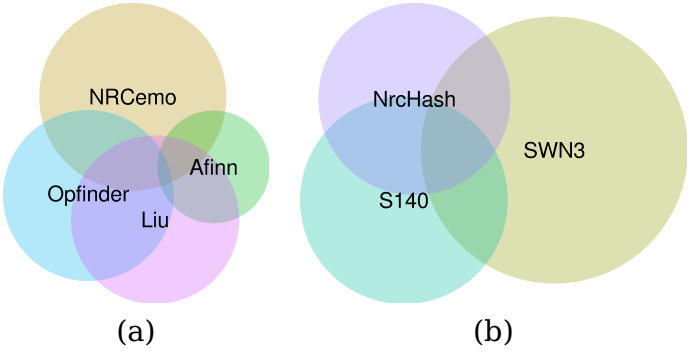

Figure 2.1: Venn diagrams of the overlap between opinion lexicons. a) lexicons created manually, and b) lexicons created automatically.

The number of words in the intersection of two lexicons is shown in Table 2.2. From the table we can see that resources created automatically, SWN3, NRC-hash, and S140Lex are much larger than resources created manually. This is intuitive, because SWN3 was created from WordNet, which is a large semantic network, and NRC-hash and S140Lex were both formed by all the different words found in their respective large collections of tweets. The overlap of the lexicons created manually is better represented in the first Venn diagram shown in Figure 2.1. We observe that Liu and OpinionFinder exhibit significant overlap. The second Venn diagram in Figure 2.1 considers lexicons created automatically. We can see that the overlap between both lexicons built from Twitter data is greater than the overlap they have with SWN3. This suggests that Twitter-made lexicons contain several expressions that are specific to Twitter.

The level of *uniqueness* of each resource is shown in Table 2.3. This value corresponds to the fraction of words of the lexicon that are not included in any

of the remaining resources. We can see that while lexicons created manually tend to have a low uniqueness, resources created automatically tend to have a substantial level of uniqueness. Nevertheless, the AFINN lexicon contains several words that are not included in other lexicons despite being the smallest lexicon created manually. This is because AFINN contains several Internet acronyms and slang words.

Lexicon	Annotation	Uniqueness	Neutrality
OpFinder	Manual	0.01	0.06
AFINN	Manual	0.19	0.00
S140Lex	Automatic	0.51	0.62
NRC-hash	Automatic	0.29	0.72
Liu	Manual	0.05	0.00
SWN3	Automatic	0.82	0.74
NRC-emotion	Manual	0.01	0.54

Table 2.3: Neutrality and uniqueness of each Lexicon. The lexicons are categorised according to the annotation mechanism.

We also studied the level of *neutrality* of each resource, shown in the second column of Table 2.3. This value corresponds to the fraction of words of the lexicon that are very likely to be neutral, and hence, are not sentiment-bearing words. The criteria for determining if a word is neutral varies from one lexicon to another. In OpinionFinder, neutral words are marked explicitly. Conversely, AFINN and Liu do not have neutral words. For the case of S140Lex and NRC-hash we consider as neutral all the words for which the absolute value of the score was less than one. In a similar manner, we consider as neutral all the words of SWN3 for which the neutral probability was one. Finally, for NRC-emotion we consider as neutral, all words that are not associated with any emotion or polarity class. Regarding the lexicons created manually we see that only NRC-emotion has a significant level of neutrality. Resources created automatically present the highest levels of neutrality. This is because they include all the words from the sources used to create them (WordNet and Twitter). Consequently, as WordNet and Twitter contain a great diversity of words or expressions, it is expected for their derived lexicons to contain many non sentiment-bearing words.

In addition to comparing the words contained in the lexicons, we also compared the level of agreement between their positive and negative polarities. We extracted the 609 words that intersect all the different lexicons. Afterwards, all the sentiment values assigned by each lexicon were converted to polarity categories positive and negative. For lexicons with numerical scores,

AFINN, S140Lex, and NRC-hash, we used the score's sign to determine positive and negative labels. For SW3N we use the sign of the difference between positive and negative probabilities. For NRC-emotion we used the polarity dimensions provided by the lexicon. The agreement between two lexicons is calculated as the fraction of words from the global intersection where both lexicons assigned the same polarity label to the word. The levels of agreement between all lexicons are shown in Table 2.4. From the Table we see that automatically created lexicons show a high level of disagreement with the human-made lexicons and even greater levels of disagreement between each other, e.g., SWN3 with S140Lex, and SWN3 with NRC-hash. That means that these larger resources tend to provide noisy information, which is far from being consolidated. On the other hand, human judgements are more likely to agree with each other.

Agreement	OpFinder	AFINN	S140Lex	NRC-hash	Liu	SWN3	NRC-emotion
OpFinder	1	×	×	×	×	×	×
AFINN	0.99	1	×	×	×	×	×
S140Lex	0.82	0.82	1	×	×	×	×
NRC-hash	0.79	0.79	0.72	1	×	×	×
Liu	0.99	0.99	0.82	0.79	1	×	×
SWN3	0.85	0.76	0.66	0.64	0.84	1	×
NRC-emotion	0.99	0.99	0.84	0.82	0.99	0.86	1

Table 2.4: Agreement of lexicons.

From the 609 words that are contained in all the lexicons, 292 present at least one disagreement between two different lexicons. A group of words presenting disagreements along the different types of lexicons is presented in Table 2.5. We can see that words such as "excuse", "joke", and "stunned" may be used to express either positive and negative opinions, depending on the context. Considering that it is very hard to associate all the words with a single polarity class, we think that emotion tags more accurately explain the diversity of sentiment states triggered by these kinds of words. For instance, the word "stunned", which is associated with both positive and negative polarities, is also associated with surprise and fear emotions. As this word would more likely be negative in the context of "fear", it would also be more likely to be "positive" in the context of "surprise".

All the insights revealed from this analysis indicate that lexical resources for sentiment analysis complement each other, providing different sentiment information and also exhibit different levels of noise. A previous study on the

word	OpFinder	AFINN	S140Lex	NRC-hash	LiuLex	SWN3	NRC-emotion
excuse	pos	-1	0.34	-1.08	neg	0.00	neg
futile	neg	2	0.05	0.07	neg	-0.50	sad
irresponsible	neg	2	-1.11	-1.87	neg	0.50	neg
joke	pos	2	-0.02	-1.50	neg	0.32	neg
stunned	pos	-2	-0.14	0.38	pos	-0.31	neg, sur, fea

Table 2.5: Sentiment values for different words. The scores in the SWN3 column correspond to the difference between the positive and negative probabilities assigned by SWN3 to the word.

relationship between different opinion lexicons was presented as a tutorial by Christopher Potts at the Sentiment Analysis Symposium[14].

2.5 Analysis of Aggregated Social Media Opinions

As has been discussed in this Chapter, sentiment analysis applied to social media is a blossoming field of research. Several applications have been developed using sentiment analysis in different contexts, including marketing studies (Jansen et al., 2009) and social sciences studies (Dodds and Danforth, 2010). In this section we review works conducting aggregated analysis of opinionated data aimed at understanding temporal aspects of opinions and further, at predicting future events from social media.

2.5.1 Temporal Aspects of Opinions

The temporal study of opinions is concerned with evaluating the *aggregated sentiment* of a target population for a certain time period. In this direction, a tool called *Moodviews* was proposed in (Mishne and de Rijke, 2006) to analyse temporal change of sentiment from LiveJournal[15] blogs. Users of LiveJournal can label their posts with mood tags from a list of 132 categories, e.g., amused or angry. Moodviews tracks the stream of these tags and allows the visualisation of mood changes through time.

A temporal analysis of sentiment events using a method based on Conditional Random Field (CRF) was performed in (Das, Kolya, Ekbal and Bandyopadhyay, 2011). The authors included sentiment features of events in order to identify temporal relations between different events from text sources.

In (Mishne and de Rijke, 2006), it was shown that opinions exhibit a certain

[14] http://sentiment.christopherpotts.net/lexicons.html
[15] http://www.livejournal.com/

degree of seasonality in Twitter. The authors found that people tend to awake in a good mood that decays during the day and that people are happier on weekends than weekdays.

The online detection of temporal changes in public opinion is studied in (Akcora, Bayir, Demirbas and Ferhatosmanoglu, 2010). The authors state that a breakpoint in public opinion is formed by a change in both the emotion pattern and the word pattern of Twitter messages. The tweets on a certain topic TT in a time period T are used to create both a vector of sentiment dimensions $\overrightarrow{v}$ and a set formed by the words within the tweets $\mathrm{Set}(T_i)$, where the vector represents the sentiment pattern, and the set represents the word pattern. Similarity measures are used to compare the word and the sentiment patterns between different periods of tweets. A period T_n must satisfy the following conditions in the two patterns in order to be considered as a breakpoint:

$$\mathrm{Sim}(T_{n-1}, T_n) < \mathrm{Sim}(T_{n-2}, T_{n-1}) \tag{2.3}$$

$$\mathrm{Sim}(T_{n-1}, T_n) < \mathrm{Sim}(T_n, T_{n+1}). \tag{2.4}$$

In (O'Connor et al., 2010), two mechanisms for measuring public opinion were compared: polls and opinions extracted from Twitter data. The authors compared several surveys on consumer confidence and political opinion, like the Gallup Organization's Economic Confidence Index and the Index of Consumer Sentiment (ICS), with sentiment ratio time series. The series were created from Twitter messages by counting positive and negative words from an opinion lexicon and computing the following expression:

$$x_t = \frac{\mathrm{count}_t(\text{pos. word} \wedge \text{topic word})}{\mathrm{count}_t(\text{neg. word} \wedge \text{topic word})} \tag{2.5}$$

Furthermore, the series were smoothed using a moving average in order to reduce volatility and derive a more consistent signal. The correlation analysis between the polls and the sentiment ratio series showed that the sentiment series are able to capture broad trends in the survey data. Nevertheless, the results showed great variation among different datasets. For example, while a high correlation between the sentiment ratio series and the index of Presidential Job Approval was observed, the correlation between the sentiment series and the pre-electoral polls for the U.S. 2008 Presidential elections was non-significant.

Opinion time series created from Twitter data were also explored in (Logunov and Panchenko, 2011). The authors sampled around $40,000,000$ tweets

in a period of 510 days using the Twitter streaming API. The sentiment evaluation of the tweets was conducted according to four emotion states: happy, sad, very happy, and very sad. A number of emoticons was mapped to each emotion state, assuming that a tweet with one of these emoticons will be associated with the corresponding emotion state. In this manner, emotion-oriented time series were calculated according to the proportion of tweets associated to each emotion state over the total number of messages in a day. The resulting time series were analysed focusing on the study of seasonal and volatility patterns. The experimental results indicated the presence of significant weekly seasonality factors and also the presence of conditional heteroskedasticity (or volatility) in the time series.

2.5.2 Predictions using Social Media

As stated in (Yu and Kak, 2012), not all topics or subjects are well suited for making predictions from social media. First of all, the topic must be related to a human event, which means that social media cannot be used to predict events whose development is independent of human actions (e.g., an eclipse or an earthquake). Secondly, there are some topics in which it is considered impolite to express opinions with a certain orientation. Therefore, the topics need to be freely discussed by people in public, otherwise the content will be biased. In the following, we present some work on predictions based on social media.

Stock Market

Stock market prediction has been traditionally addressed through the random walk theory and the Efficient Market Hypothesis (EMH). This approach states that stock market prices reflect all publicly available information and adjust rapidly to the arrival of new information. Moreover, due to the fact that the arrival of information is unpredictable, stock prices follow a random walk process and cannot be accurately predicted in the long term. However, there is work on using social media data and opinion mining methods as a proxy for predicting stock market prices. In (De Choudhury, Sundaram, John and Seligmann, 2008) the communication dynamics in the blogosphere were studied, showing a considerable correlation between social data and stock market activity. An SVM regressor was trained using contextual properties of communications for a particular company as features and the stock market movement

of the company as target variable. Some of the features considered were: the number of posts, the number of comments, the length and response time of comments, among others. An accuracy of 78% was obtained for predicting the magnitude of movement and 87% for the direction of movement. In (Bollen et al., 2011), it was investigated whether public moods extracted from Twitter data can be used to predict the stock market. Two methods were used to create mood time series from a collection of $9,853,498$ tweets from February 28 to December 19th. The first method uses the *OpinionFinder* lexicon to create a positive vs. negative daily time series, and the second one uses Google-Profile of Mood States (GPOMS) to create a six-dimensional daily time series based on the following mood states: Calm, Alert, Sure, Vital, Kind, and Happy. In order to asses the ability of these time series to predict stock market changes, they compared them with the Dow Jones Industrial Average (DJIA) using the econometric technique of Granger causality analysis (Granger, 1969). The results indicate that prediction of the stock market can be significantly improved when mood dimensions Calm and Happiness are considered. The other mood dimensions were not predictive.

Movie Box-Office Revenue

"Movie box-office revenue", is a concept used to describe how successful a movie is. There are a number of works that use social media to predict movie performance, e.g. (Asur and Huberman, 2010; Liu, Huang, An and Yu, 2007; Mishne and Glance, 2006). According to (Yu and Kak, 2012), there are several reasons why predicting movie box-office revenue is a good subject of research. The first reason is the availability of large volumes of data about movies and the easy access to them. *The Internet Movie Database*[16] (IMDB) provides box-office indicators such as the gross income of released movies. Furthermore, social media users that post about a movie before its release date are very likely to end up watching it. Therefore, there is a clear correlation between social media and movie box-office. For example, in (Asur and Huberman, 2010), authors found more than $100,000$ tweets for each monitored movie. In that work, tweets were used to forecast box-office revenues for movies using properties such as the rate at which tweets are created and sentiment indicators. An Autoregressive Sentiment Aware model (ARSA) to predict box office performance from blogs was proposed in (Liu et al., 2007). The model assumes that each blog document is generated by a number of

[16] http://www.imdb.com/

hidden sentiment factors which are estimated using the Expectation Maximisation algorithm (EM). Then, movie box revenues are predicted by combining an autoregressive model of past revenues with sentiment factors extracted from blogs.

Politics

The result of political elections has been traditionally predicted through public opinion surveys such as telephone surveys or polls. A limitation of polls is that they are expensive and need to be conducted periodically in order to track voting intentions over time. Due to this, predicting elections with social media has become an active area of research.

According to (Gayo-Avello, 2013), there are essentially two approaches for inferring election votes from Twitter data: 1) by counting tweets, and 2) by relying on sentiment analysis. The rationale behind counting tweets is quite simple: the larger the number of tweets mentioning the target candidate, the larger the vote rate (Tumasjan, Sprenger, Sandner and Welpe, 2010). On the other hand, approaches based on sentiment analysis rely on polarity lexicons (O'Connor et al., 2010) and supervised learning (Ceron, Curini, Iacus and Porro, 2014) for calculated aggregated sentiment indexes about the candidates. Metaxas, Mustafaraj and Gayo-avello (2011) found that sentiment analysis outperforms counting tweets for vote intention estimation.

There is no clear consensus about the predictive performance of election predictions based on social media and opinion mining. For example, Tumasjan et al. (2010) argue that the predictive power of this approach is "close to traditional election polls". In (Ceron et al., 2014) results were worse than traditional polls in terms of Mean Average Precision (MAE) but still reasonable. In (Gayo-Avello, 2011), it is stated that this predictive performance is greatly exaggerated. Furthermore, there are cases in which different social media predictions for the same event give conflicting results. While the authors of (Tumasjan et al., 2010) claim that German elections of 2009 could have been predicted using Twitter, the opposite is stated in (Jungherr, Jurgens and Schoen, 2011).

2.6 Discussion

In this chapter, we provided a broad review of the field of sentiment analysis. Several methods for polarity classification were described as well as methods

for polarity lexicon induction. We provided new evidence concerning the label sparsity and sentiment-drift problems in sentiment analysis. We also studied the properties of popular lexical resources and showed real-world applications of sentiment analysis of social media, such as such stock market prediction and the prediction of political elections.

Considering methods for polarity classification of documents, sentences, and tweets, we observed how lexical knowledge can aid this task, especially when labelled data is scarce. Moreover, we observed that information about the sentiment of messages can be very useful for inducing the polarity of words. This suggests the existence of a sentiment-interdependence relation between words and documents. We hypothesise that given the short length of tweets, this interdependence is statistically more significant in tweets than in other types of documents and we will exploit it in this thesis for both polarity lexicon induction and message-level polarity classification tasks. The interdependence relation is described by the following two statements:

1. The polarity of words is determined by the polarity of the tweets in which they occur.

2. The polarity of tweets is determined by the polarity of their words.

The first lexicon induction method in this thesis, described in Chapter 3, which exploits word sentiment associations, is based on the first part of the relation, as it relies on automatically labelled tweets for performing the induction of a Twitter-specific opinion lexicon.

The tweet-centroid model we propose for both lexicon induction and distant supervision in Chapters 4 and 5 is a unified representation that allows the bidirectional transfer of sentiment classifiers between words and tweets. The main benefit of this approach is that it only requires labelled data in one of the two domains (words or messages) for transferring sentiment knowledge into the other one. Moreover, the model is capable of performing the induction of a polarity lexicon from unlabelled tweets based on a seed lexicon. This is a crucial property of the model because it means that is can be used for building domain-specific lexicons in domains where emoticons are rarely used to express both positive and negative sentiment, such as politics.

Finally, the ASA method proposed in Chapter 6 is also grounded in the interdependence relation, as it exploits prior lexical knowledge and unlabelled data for creating synthetic polarity data by sampling and averaging multiple

tweets without requiring any labelled tweets. ASA works on the whole message rather than being entity oriented as the method in (Zhang et al., 2011). Moreover, ASA can be used for creating training data of any size and distribution of labels and hence may be useful for dealing with the class imbalance problem reported in (Li et al., 2011).

Chapter 3

Word-sentiment Associations for Lexicon Induction

In this chapter, we propose a method for opinion lexicon induction for the language used in Twitter. It applies supervised learning using word-level sentiment associations. Taking SentiWordNet (Baccianella et al., 2010) as inspiration, each word in our expanded lexicon has a probability distribution, describing how positive, negative, and neutral it is. Additionally, all the entries of the lexicon are associated with a corresponding part-of-speech tag. Estimating the sentiment distribution of POS-tagged words is useful for the following reasons:

1. A word can present certain levels of intensity (Thelwall et al., 2012) for a specific sentiment category, e.g., the word *awesome* is more positive than the word *adequate*. The estimated probabilities can be used to represent these levels of intensity. These probabilities provide a probabilistic interpretation of the underlying sentiment intensities conveyed by a word and can be used as prior knowledge in Bayesian models for sentiment inference (Lin and He, 2009). In contrast, scores obtained by unsupervised methods such as point-wise-mutual information semantic orientation (PMI-SO) (Turney, 2002) lack a probabilistic interpretation.

2. The neutrality score provided by the lexicon is useful for discarding non-opinion words in text-level polarity classification tasks. This can easily be done by discarding words classified as neutral. Note that unsupervised lexicon expansion techniques such as PMI-SO (Turney, 2002) provide a single numerical score for each word. Therefore, there would be a need to empirically establish a threshold on this score for neutrality detection.

3. Homographs, which are words that share the same spelling but have different meanings, should have different lexicon entries for each different

meaning. By using POS-tagged words, homographs with different POS-tags will be disambiguated (Wilks and Stevenson, 1998). For instance, the word *fine* will receive different sentiment scores when used as an adjective (e.g., *I'm fine thanks*) and as a common noun (e.g., *I got a parking fine because I displayed the ticket upside down*).

This is not the first work exploring these properties for lexicon expansion. Sentiment intensities were described with probabilities in (Baccianella et al., 2010), and the disambiguation of the sentiment of words based on POS tags was studied in (Taboada et al., 2011). However, this is the first time that these properties are explored for the informal language used in Twitter.

Our expanded lexicon is built by training a word-level sentiment classifier for the words occurring in a corpus of polarity-annotated tweets. The training words are labelled using a seed lexicon of positive, negative, and neutral words. This lexicon is taken from the union of four different hand-made lexicons after discarding all polarity clashes from the intersection. The expanded words are obtained after deploying the trained classifier on the remaining unlabelled words from the corpus of tweets that are not included in the seed lexicon.

All the words from the polarity-annotated corpus of tweets are represented by features that capture morphological and sentiment information of the word in its context. The morphological information is captured by including the POS tag of the word as a nominal attribute, and the sentiment information is captured by calculating association values between the word and the polarity labels of the tweets in which it occurs.

We calculate two types of word-level sentiment associations: PMI-SO (Turney, 2002), which is based on the point-wise mutual information (PMI) between a word and tweet-level polarity classes, and stochastic gradient descent semantic orientation (SGD-SO), which is based on incrementally learning a linear association between words and the sentiment of the tweets in which they occur.

To avoid the high costs of manually annotating tweets into polarity classes for calculating the word-level sentiment associations, we rely on two heuristics for automatically obtaining polarity-annotated tweets: the *emoticon-annotation approach*, introduced in Chapter 1, and *model transfer*. In the first approach, only tweets with positive or negative emoticons are considered and labelled according to the polarity indicated by the emoticon. The main drawbacks of this approach is that the removal of tweets without emoticons may cause a

loss of valuable words that do not co-occur with emoticons, and that there are domains in which emoticons are rarely used to express positive or negative opinions.

To overcome these limitations, we pursue a model transfer approach by training a probabilistic message-level classifier from a corpus of emoticon-annotated tweets and using it to label a target corpus of unlabelled tweets with a probability distribution of positive and negative sentiment. Note that the model transfer produces soft sentiment labels, in contrast to the hard labels provided by emoticons. We study how to compute our word-level sentiment association attributes from tweets annotated with both hard and soft labels.

We test our word-level sentiment classification approach on words obtained from different collections of automatically labelled tweets. The results indicate that our supervised framework outperforms using PMI-SO by itself when the detection of neutral words is considered. We also evaluate the usefulness of the expanded lexicon for classifying entire tweets into polarity classes, showing significant improvement in performance compared to the original lexicon.

This chapter is organised as follows. In Section 3.1, we describe the proposed method in detail. In Section 3.2, we present the experiments we conducted to evaluate the proposed approach and discuss results. The main findings are discussed in Section 3.3.

3.1 Proposed Method

In this section we describe the proposed method for opinion lexicon expansion from automatically annotated tweets. The proposed process is illustrated in Figure 3.1, and can be summarised in the following steps:

1. Collect tweets from the target domain and the time period for which the lexicon needs to be expanded.

2. If the target collection has a significant number of positive and negative emoticons, label it using the emoticon-based annotation approach. Otherwise, collect tweets with positive and negative emoticons from a general domain and use it to label the target collection with the model transfer approach discusses below.

3. Tag all the words from the target collection using a part-of-speech tagger.

4. Calculate word-level features for all tagged words.

5. Label these words with a sentiment that matches an existing hand-made polarity lexicon.

6. Train a word-level classifier using the word-level features and the word labels from the seed lexicon.

7. Use the trained classifier to estimate the polarity distribution of the remaining unlabelled words.

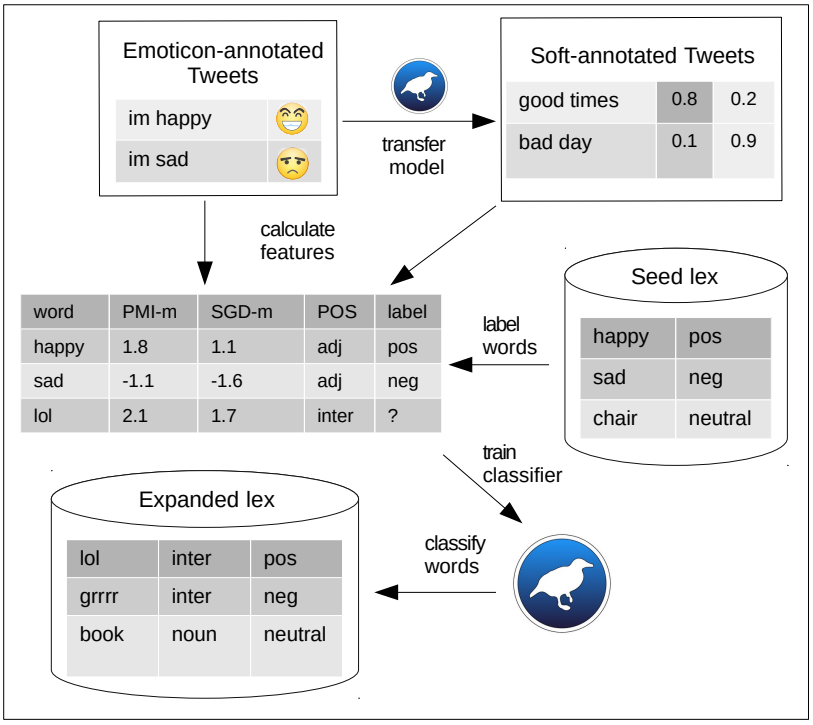

Figure 3.1: Twitter-lexicon induction process. The bird represents the Weka machine learning software.

The key parts of the methodology are described in the following subsections. The mechanisms studied to automatically create collections of labelled tweets are described in Section 3.1.1. The proposed word-level attributes are described in Section 3.1.2. The seed lexicon used to label the training words is described in Section 3.1.3.

3.1.1 Automatically-annotated Tweets

The proposed method for lexicon induction requires a collection of tweets with two properties: 1) the tweets must be labelled according to positive and negative polarity classes, and 2) they must be sorted in chronological order.

The first property is necessary for calculating our word-level features based on associations between words and the polarity of the tweets in which they occur (Section 3.1.2).

The second property is also necessary for calculating our word-level features, because they exploit how the associations between words and tweet-level polarities evolve over time.

The limitation of depending on polarity-annotated tweets is that the process of annotating tweets into polarity classes is labor-intensive and time-consuming. We tackle this problem by employing two automatic heuristics for data annotation: emoticon-based annotation and model transfer annotation.

Emoticon-based Annotation

In the emoticon-based annotation approach, tweets exhibiting positive :) and negative :(emoticons are labelled according to the emoticon's polarity (Go et al., 2009). Afterwards, the emoticon used to label the tweet is removed from the content. The emoticon-based labels are denoted by the letter y, and are assumed to be in $\{+1, -1\}$, corresponding to positively and negatively labelled tweets, respectively.

In the same way as in (Go et al., 2009), the attribute space is reduced by replacing sequences of letters occurring more than two times in a row with two occurrences of them (e.g., huuungry is reduced to huungry, loooove to loove), and replacing user mentions and URLs with the generic tokens "USER" and "URL", respectively.

We consider two collections of tweets covering multiple topics for building our datasets: The Stanford Sentiment corpus (STS) (Go et al., 2009), and The Edinburgh corpus (ED) (Petrović et al., 2010). These collections were gathered from two Twitter APIs: the search API[1], which allows the submission of queries composed of key terms, and the streaming API[2], from which a real-time sample of public posts can be retrieved.

The STS corpus is an emoticon-annotated collection created by periodically sending queries :) and :(to the Twitter search API between April 6th 2009 to June 25th 2009. The ED corpus is a general purpose collection of 97 million unlabelled tweets in multiple languages. It was collected with the Twitter streaming API between November 11th 2009 and February 1st 2010. We applied the emoticon-based annotation approach to the tweets written in English

[1] https://dev.twitter.com/rest/public/search
[2] https://dev.twitter.com/streaming/overview

from this collection. We refer to this emoticon-annotated collection as ED.EM. The number of tweets for each polarity class in the two emoticon-annotated corpora is given in Table 3.1. We can observe that when using the streaming API (ED), positive emoticons occur much more frequently than negative ones.

	ED.EM	STS
Positive	$1,813,705$	$800,000$
Negative	$324,917$	$800,000$
Total	$2,138,622$	$1,600,000$

Table 3.1: Emoticon-annotated datasets.

As was discussed in Chapter 1, the shortcoming of the emoticon-annotation approach is that it discards a large amount of potentially valuable information and is useless in domains where emoticons are infrequently used to express sentiment.

Model Transfer Annotation

The model transfer approach enables the extraction of opinion words from any collection of tweets. It tackles the problems of the emoticon-annotation approach by employing a self-training approach. The idea is to train a probabilistic message-level classifier from a source corpus $\mathcal{C}_s$ of emoticon-annotated tweets and then use it to classify a target-corpus of unlabelled data $\mathcal{C}_t$. We use an L_2-regularised logistic regression model with unigrams as attributes for training the classifier and labelling the target collection with soft labels. The soft labels of a tweet $d \in \mathcal{C}_t$ are denoted as $pos(d)$ and $neg(d)$, and represent a probability distribution of positive and negative sentiment (i.e., $1 - pos(d) = neg(d)$).

An important difference between the model transfer and emoticon-annotation approach is the nature of the sentiment labels they produce. The emoticon-based annotation approach produces hard labels $y \in \{+1, -1\}$; the model transfer produces soft ones $pos(d), neg(d) \in [0, 1]$.

The soft labels can easily be converted into hard ones by impossing a threshold λ and discarding tweets for which the classifier is not confident enough in its prediction:

$$y(d) = \begin{cases} 1 & \text{if } pos(d) \geq \lambda \\ -1 & \text{if } neg(d) \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The tweets for which $y(d) = 0$ are then discarded. It is worth pointing out that the removal of these tweets may lead to the loss of valuable data, and that λ is a tuning parameter that needs to be adjusted with possible values between 0.5 and 1. An alternative approach is to use the soft labels directly. We will study strategies for extracting word-level attributes for both hard and soft labels in Section 3.1.2.

It is noteworthy to mention that in contrast to the emoticon-annotated source corpus, which is intentionally biased towards positive and negative tweets, the target collection may contain a substantial amount of neutral data or even tweets with mixed positive and negative sentiment. It is unclear how a classifier trained to discriminate between positive and negative tweets will behave when deployed on tweets that have a different sentiment class such as neutral or mixed. This might be a shortcoming of the model transfer approach. However, it is plausible that neutral tweets or tweets with mixed sentiment will be located close to the decision boundary of the classifier trained from positive and negative emoticons. Therefore, we can expect that the soft labels obtained with logistic regression for these types of tweets will have similar probabilities for both positive and negative classes and will be discarded when setting a sufficiently high value of λ.

The data we use for our model transfer experiments is obtained using the STS corpus as the source collection, and a sample of 10 million tweets from ED as the target collection. The classifier we use is an L_2-regularised logistic regression model with the regularisation parameter C set to 1, generated using LIBLINEAR[3]. We refer to this corpus of tweets annotated with soft labels as ED.SL. The average values for the positive and negative soft labels in ED.SL are 0.64 and 0.36 respectively. We also convert this soft-annotated corpus into multiple hard-annotated datasets using different thresholds values. We refer to these collections as ED.Tα, where α is the value of the threshold. The number of positive and negative tweets in the resulting datasets is shown in Table 3.2. Note that the higher the value of α, the more tweets are discarded.

[3] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

Dataset	ED.T06	ED.T07	ED.T08	ED.T09
Positive	$6,279,007$	$5,164,387$	$3,761,683$	$2,030,418$
Negative	$2,182,249$	$1,609,195$	$1,090,086$	$586,441$
Total	$8,461,256$	$6,773,582$	$4,851,769$	$2,616,859$

Table 3.2: Model transfer datasets with different threshold values.

3.1.2 Word-level Features

In this subsection, we provide a detailed description of the word-level features used for classifying words from a corpus of polarity-annotated tweets into positive, negative, and neutral classes.

We preprocess the given corpus before calculating the features. All the tweets are lowercased, tokenised and POS-tagged. We use the TweetNLP library (Gimpel, Schneider, O'Connor, Das, Mills, Eisenstein, Heilman, Yogatama, Flanigan and Smith, 2011), which provides a tokeniser and a tagger specifically for the language used in Twitter. We prepend a POS-tag prefix to each word in order to differentiate homographs exhibiting different POS-tags.

The first feature is a nominal attribute corresponding to the POS tag of the word in its context. This feature provides morphological information of the word. There is empirical evidence that subjective and objective texts have different distributions of POS tags (Pak and Paroubek, 2010). According to (Zhou et al., 2014), non-neutral words are more likely to exhibit the following POS tags in Twitter: noun, adjective, verb, adverb, abbreviation, emoticon and interjection. These findings suggest that POS tags may provide useful information for discriminating between neutral and non-neutral words.

The remaining features aim to capture the association between the POS-tagged word and sentiment. We sort the tweets from the collection chronologically and create two semantic orientation time series for each word: the SGD-SO series, and the PMI-SO series. These time series are designed to capture the evolution of the relationship between a word and the sentiment that it expresses. The way in which these series are calculated varies according to the nature of the sentiment labels of the tweets. As was described in Section 3.1.1, there are two types of message-level sentiment labels we can obtain from our methods for annotating tweets automatically: hard labels (positive or negative), and soft labels ($pos(d), neg(d)$). The hard labels are obtained when using the emoticon-annotation approach and the soft ones from the transfer model one. It is also possible to obtain hard-labels from the model transfer

approach by applying a threshold parameter λ.

Features Calculated from Hard Labels

The first semantic orientation time series is calculated by incrementally training a linear model using stochastic gradient descent (SGD-SO). The weights of this model correspond to POS-tagged words and are updated in an incremental fashion. For the hard-labelled data ($y \in \{+1, -1\}$), we incrementally train a support vector machine (Zhang, 2004) by optimising the hinge loss function with an L_2 penalty and a learning rate equal to 0.1:

$$\frac{\lambda}{2}||w||^2 + \sum[1 - y(\mathbf{x}\mathbf{w} + b)]_+. \tag{3.2}$$

The variables $\mathbf{w}$, b, and λ correspond to the weight vector, the bias, and the regularisation parameter, respectively. The regularisation parameter was set to 0.0001. The model's weights determine how strongly the presence of a word influences the prediction of negative and positive classes (Bifet and Frank, 2010). The SGD-SO time series is created by applying this learning process to a collection of labelled tweets and storing the word's coefficients in different time windows. We use time windows of $1,000$ examples.

The second time series corresponds to the accumulated PMI semantic orientation (PMI-SO), which is the difference between the PMI of the word with a positive sentiment and the PMI of the word with a negative sentiment (Turney, 2002):

$$\begin{aligned} \text{PMI-SO}(w) &= \text{PMI}(w, \text{pos}) - \text{PMI}(w, \text{neg}) \\ &= \log_2\left(\frac{\Pr(w, pos)}{\Pr(w) \times \Pr(pos)}\right) - \log_2\left(\frac{\Pr(w, neg)}{\Pr(w) \times \Pr(neg)}\right) \\ &= \log_2\left(\frac{\Pr(w, pos) \times \Pr(neg)}{\Pr(pos) \times \Pr(w, neg)}\right) \end{aligned} \tag{3.3}$$

Let *count* be a function that counts the number of times that a word or a sentiment label has been observed up to and including a certain time period. For hard-labelled tweets, we calculate the PMI-SO score for each POS-tagged word according to the following expression:

$$\text{PMI-SO}(w) = log_2\left(\frac{\text{count}(w \wedge y = 1) \times \text{count}(y = -1)}{\text{count}(y = 1) \times \text{count}(w \wedge y = -1)}\right) \tag{3.4}$$

We use time windows of $1,000$ examples and the Laplace correction to avoid

the zero-frequency problem.

Feature	Description
mean	The mean of the time series.
trunc.mean	The truncated mean of the time series.
median	The median of the time series.
last.element	The last observation of the time series.
sd	The standard deviation of the time series.
iqr	The inter-quartile range.
sg	The fraction of times the time series changes its sign.
sg.diff	The sg value applied to the differenced time series ($X_t - X_{t-1}$).

Table 3.3: Time series features.

We use the time series to extract features that are used to train our world-level polarity classifier. These features summarise location and dispersion properties of the time series, and are listed in Table 3.3. The location-oriented features *mean*, *trunc.mean* and *median* measure the central tendency of the time series. The dispersion oriented features *sd*, *iqr*, *sg*, and *sg.diff* measure the variability of the time series. The feature *last.element* corresponds to the last value observed in the time series. This attribute would be equivalent to the traditional PMI semantic orientation measure for the PMI-SO time series calculated from hard labels.

Features Calculated from Soft Labels

In the scenario of tweets with soft labels, the PMI-SO and SGD-SO time series are calculated in a different way.

For the SGD-SO time series we use an L_2 regularised squared loss function. Let z be a real value that corresponds to the log odds of the positive and negative sentiment labels of a given tweet: $z = \log_2(\frac{pos(d)}{neg(d)})$, and let the variables $\mathbf{w}$, b, and λ be analogous to the ones from the hinge loss. The squared loss function is defined as follows:

$$\frac{\lambda}{2}||w||^2 + \sum(z - (\mathbf{xw} + b))^2. \tag{3.5}$$

The PMI-SO time series is calculated using soft counts. Let C be the set of tweets seen so far and $C(w)$ be the tweets from C in which the word w is

observed. Then, the soft version of PMI-SO is calculated as follows:

$$\text{PMI-SO}'(w) = log_2 \left(\frac{\sum_{d \in C(w)} \text{pos}(d) \times \sum_{d \in C} \text{neg}(d)}{\sum_{d \in C} \text{pos}(d) \times \sum_{d \in C(w)} \text{neg}(d)} \right) \tag{3.6}$$

We calculate the same features (Table 3.3) from the soft versions of the SGD-SO and PMI-SO time series as the ones calculated from their corresponding hard versions.

3.1.3 Ground-Truth Word Polarities

In this subsection, we describe the seed lexicon used to label the training words for our word sentiment classifier. In order to create an expanded lexicon similar to SentiWordNet, we require a seed lexicon of words manually labelled according to mutually exclusive positive, negative, and neutral sentiment classes. We create it by fusing the following manually created lexical resources that were described in Chapter 2:

- *MPQA Subjectivity Lexicon*: We consider positive, negative, and neutral words from this lexicon.
- *Bing Liu*: We consider positive and negative entries from this lexicon.
- *AFINN*: We tagged words with negative and positive scores from this lexicon to negative and positive classes respectively.
- *NRC emotion Lexicon*: We consider positive, negative, and neutral words from this lexicon, and the words associated with both positive and negative categories are discarded. The neutral words correspond to the words that are not associated with any emotion or polarity category.

As discussed in Chapter 2, different lexical resources may assign different categories to the same word. In order to create mutually exclusive polarity classes and to reduce the noise in our training data, we discard all words for which a polarity clash is observed. A polarity clash is a word that receives two or more different tags in the union of lexicons.

The number of words for the different polarity classes in the different lexicons is displayed in Table 3.4.

The total number of clashes is 1074. We observe that this number is higher than the one observed in Chapter 2 when the agreement between lexicons was analysed. In that experiment, manually-annotated lexicons exhibited high

	Positive	Negative	Neutral
AFINN	564	964	0
Bing Liu	2003	4782	0
MPQA	2295	4148	424
NRC-Emo	2312	3324	7714
Seed Lexicon	3730	6368	7088

Table 3.4: Lexicon Statistics.

agreement for positive and negative classes. However, we observe here that including a neutral label produces a substantial increase in the number of clashes. This high number of clashes found among different hand-made lexicons indicates two things: 1) Different human annotators can disagree when tagging a word to polarity classes, and 2) there are words that can belong to more than one sentiment class. Hence, we can say that word-level polarity classification is a hard and subjective problem.

3.2 Evaluation

In this section, we conduct an experimental evaluation of the proposed model for Twitter opinion lexicon expansion. The evaluation is divided into four parts. In the first part we conduct an exploratory analysis of word-level features calculated from real Twitter data. In the second part we evaluate the word-level classifiers. In the third part we perform lexicon expansion using the trained classifiers and study the expanded resources. In the fourth part we conduct an extrinsic evaluation by using the expanded words for message-level polarity classification of tweets.

3.2.1 Exploratory Analysis

In this subsection, we explore the proposed time series described in Section 3.1.2 and the features extracted from them with the aim of observing how these variables correlate with the sentiment of words. The time series are calculated for the most frequent $10,000$ POS-tagged words found in each of our two emoticon-annotated datasets (STS and ED.EM) using MOA[4], a data stream mining framework.

Figure 3.2 shows the resulting semantic orientation time series *SGD-SO* and *PMI-SO* for a sample of words in the ED.EM dataset. We can observe that the

[4]http://moa.cs.waikato.ac.nz/

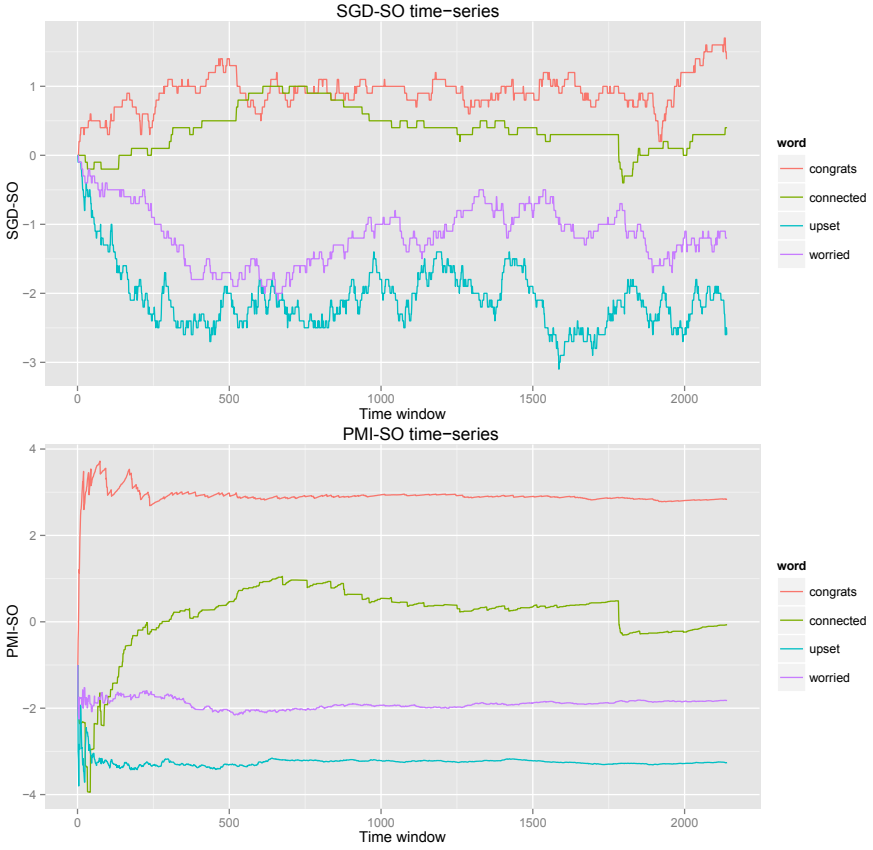

Figure 3.2: Word-level time series.

positive and neutral words *congrats* and *connected* exhibit greater PMI-SO and SGD-SO scores than the negative words *worried* and *upset*. This suggests a correspondence between time series values and word polarities. We can also see that the PMI-SO time series are much more stable than the SGD-SO ones. We believe that PMI-SO is more stable than SGD-SO because, as shown in Equation 3.4, PMI-SO treats each word independently from all other words given the sentiment class. In contrast, SGD-SO scores are updated according to the SGD-SO learning rule, which comes from the sub-gradient of Equation 3.2. In this rule, the coefficients are updated every time the learning SVM misclassifies an example from the stream of emoticon-labelled tweets ($y(\mathbf{xw} + b) < 1$). Therefore, the change of the weight of a particular word depends both on the sentiment label and the co-occurring words within the tweets from the training collection.

To create training and test data for learning a word classifier, all POS-tagged words matching the seed lexicon, and, thus, their corresponding time series, are labelled according to the lexicon's polarities. It is interesting to consider

how frequently positive, negative, and neutral words occur in a collection of tweets. The number of words labelled as positive, negative, and neutral for both the ED.EM and STS dataset is given in Table 3.5. As shown in the table, neutral words are the most frequent words in both datasets. Moreover, positive words are more frequent than negative ones.

	ED.EM	STS
Positive	1027	1023
Negative	806	985
Neutral	1814	1912
Total	3647	3920

Table 3.5: Word-level polarity classification datasets.

As the lexicon's entries are not POS-tagged, we assume that all possible POS tags of a word have the same polarity. However, this assumption can introduce noise in the training data. For example, the word *ill*, which is labelled as negative by the lexicon, will be labelled as negative for two different POS-tags: adjective, and nominal+verbal contraction. This word is very likely to express a negative sentiment when used as an adjective, but it is unlikely to express a negative sentiment when it refers, in a misspelled way, to the contraction *I'll*. A simple outlier removal technique to deal with this problem will be discussed in Section 3.2.3.

Once our time series are created, we extract from them the word-level features described in Section 3.1.2. The feature values obtained for some example words are given in Table 3.6. We can see that each entry has a POS-tag prefix. As expected, features from the same time series related to measures of central tendency (e.g., mean and median) exhibit similar values.

A scatterplot between attributes *sgd-so.mean* and *pmi-so.mean* for the labelled words from the STS corpus is shown in Figure 3.3. From the figure we can observe that the two variables are highly correlated. The correlation is 0.858 and 0.877 for the ED.EM and STS corpora respectively. Positive, negative, and neutral words are depicted with different colours. We can observe that negative words tend to show low values of sgd-so.mean and pmi-so.mean, and positive words tend to show the opposite. Neutral words are more spread out and hard to distinguish. This pattern can also be clearly seen in the boxplots shown in Figure 3.4.

The boxplots show that the three classes of words exhibit different statistical properties in both sgd-so.mean and pmi-so.mean. The medians of

Attribute	!-congrats	A-connected	A-upset	V-worried
sgd-so.last	1.4	0.4	-2.6	-1.2
sgd-so.mean	0.9	0.4	-2.1	-1.2
sgd-so.trunc.mean	0.9	0.4	-2.1	-1.2
sgd-so.median	0.9	0.4	-2.1	-1.2
sgd-so.sd	0.3	0.3	0.4	0.4
sgd-so.sg	0.0	0.0	0.0	0.0
sgd-so.sg.diff	0.0	0.0	0.1	0.0
sgd-so.iqr	0.2	0.3	0.5	0.6
pmi-so.last	2.8	-0.1	-3.3	-1.8
pmi-so.mean	2.9	0.1	-3.2	-1.9
pmi-so.trunc.mean	2.9	0.3	-3.3	-1.9
pmi-so.median	2.9	0.3	-3.2	-1.9
pmi-so.sd	0.2	0.8	0.1	0.1
pmi-so.sg	0.0	0.0	0.0	0.0
pmi-so.sg.diff	0.2	0.4	0.4	0.4
pmi-so.iqr	0.1	0.6	0.1	0.1
pmi-so.tag	interjection	adjective	adjective	verb
label	positive	neutral	negative	negative

Table 3.6: Word-level feature example.

the classes show an accurate correspondence with the word's polarity, i.e., $\text{median}(pos) > \text{median}(neu) > \text{median}(neg)$. It is worth pointing out that negative words exhibit the largest spread and that most of the boxplots show a substantial number of outliers. These outliers, which exhibit very high absolute values of sgd-so.mean and pmi-so.mean, correspond to words that occur with much greater frequency in tweets with a specific positive or negative polarity than in tweets with the opposite polarity. We also observe outliers exhibiting values of sgd-so.mean and pmi-so.mean with the opposite direction as the word's polarity. We attribute them to the following factors:

- Words that were wrongly labelled in the seed lexicon.
- Words that are frequently occurring in tweets with the opposite polarity by chance.
- Words whose polarity conveyed in the corpus of tweets is the different as the polarity provided by the lexicon.

3.2.2 Word-level Classification

In this subsection, we focus on the word-level classification problem. With the aim of gaining a better understanding of the problem, we study three word-level classification problems:

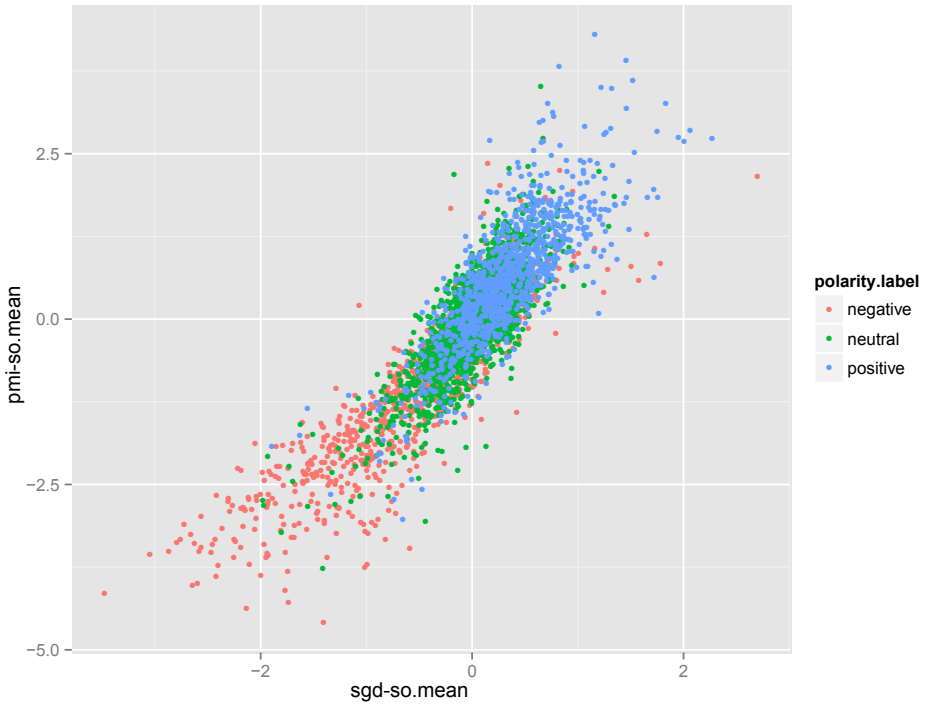

Figure 3.3: PMI-SO vs SGD-SO scatterplot.

1. *Neutrality*: Classify words as neutral (objective) or non-neutral (subjective). We label positive and negative words as non-neutral for this task.

2. *PosNeg*: Classify words as positive or negative. We remove all neutral words for this task.

3. *PosNegNeu*: Classify words as positive, negative or neutral. This is the primary classification problem we aim to solve.

In the first part of this subsection, we study word-level sentiment classification using tweets annotated with the emoticon-based annotation approach. Afterwards, we will study the same problem using tweets annotated with the model transfer approach.

Word Classification from Emoticon-annotated Tweets

We first consider the information provided by each feature with respect to the three classification tasks described above. This is done by calculating the information gain of each feature using the *R* package *FSelector*[5]. This score is normally used for decision tree learning and measures the reduction of entropy within each class after performing the best split induced by the

[5]http://cran.r-project.org/web/packages/FSelector/

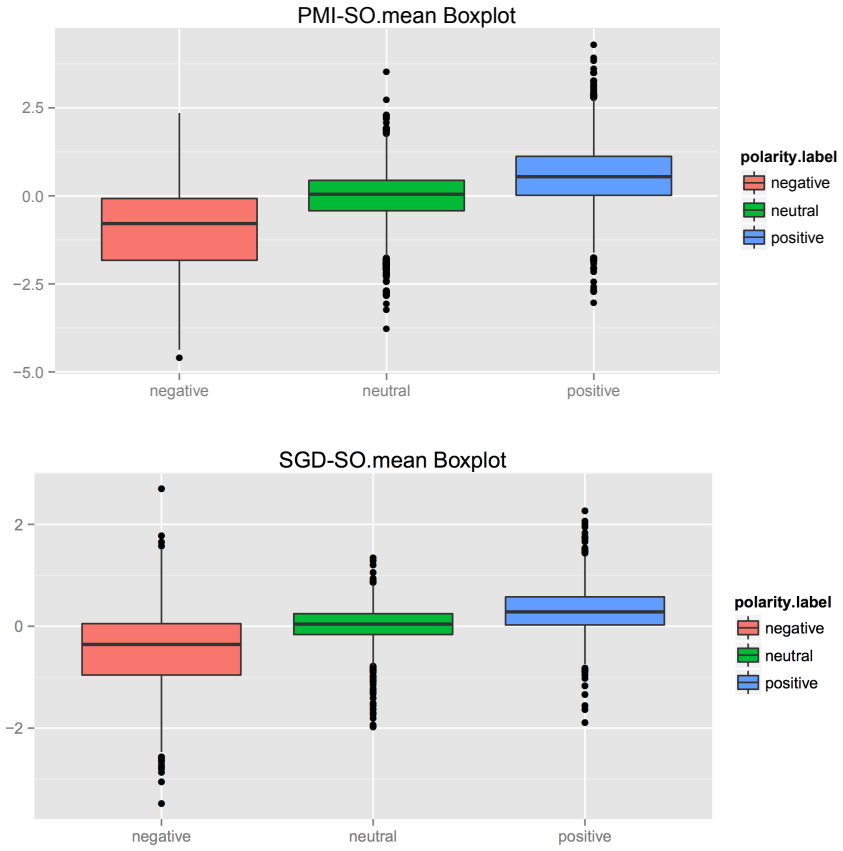

Figure 3.4: PMI-SO and SGD-SO Boxplots.

feature. The information gain obtained for the different attributes in relation to the three classification tasks is shown in Table 3.7. The attributes achieving the highest information gain per task are marked in bold.

We can observe that variables measuring the location of the PMI-SO and SGD-SO time series tend to be more informative than those measuring dispersion. Moreover, the information gain of these variables is much higher for PosNeg than for Neutrality. SGD-SO and PMI-SO are competitive measures for neutrality, but PMI-SO is better for PosNeg. An interesting insight is that features that measure the central tendency of the time series tend to be more informative than those giving the last value of the time series, especially for SGD-SO. These measures smooth the fluctuations of the SGD-SO time series. We can see that the feature *sgd-so.mean* is the best attribute for neutrality classification in both datasets. We can also see that POS tags are useful for neutrality detection, but useless for PosNeg. Therefore, we can conclude that positive and negative words have a similar distribution of POS tags.

We trained supervised classifiers for the three different classification prob-

Dataset	ED.EM			STS		
Task	Neutrality	PosNeg	PosNegNeu	Neutrality	PosNeg	PosNegNeu
pos-tag	0.062	0.017	0.071	0.068	0.016	0.076
sgd-so.mean	**0.082**	0.233	0.200	**0.104**	0.276	**0.246**
sgd-so.trunc.mean	0.079	0.237	0.201	**0.104**	0.276	0.242
sgd-so.median	0.075	0.233	0.193	0.097	0.275	0.239
sgd-so.last	0.057	0.177	0.155	0.086	0.258	0.221
sgd-so.sd	0.020	0.038	0.034	0.030	0.030	0.052
sgd-so.sg	0.029	0.000	0.030	0.049	0.017	0.062
sgd-so.sg.diff	0.000	0.000	0.008	0.005	0.000	0.000
sgd-so.iqr	0.018	0.012	0.019	0.015	0.014	0.017
pmi-so.mean	0.079	0.283	**0.219**	0.081	**0.301**	0.232
pmi-so.trunc.mean	0.077	**0.284**	0.215	0.079	0.300	0.229
pmi-so.median	0.077	0.281	0.215	0.076	0.300	0.228
pmi-so.last	0.069	0.279	0.211	0.084	0.300	0.240
pmi-so.sd	0.000	0.015	0.008	0.000	0.012	0.007
pmi-so.sg	0.013	0.216	0.126	0.019	0.239	0.142
pmi-so.sg.diff	0.000	0.012	0.009	0.000	0.000	0.000
pmi-so.iqr	0.000	0.000	0.000	0.000	0.008	0.000

Table 3.7: Information gain values. Best result per column is given in bold.

lems using both emoticon-annotated datasets, STS and ED.EM. The classification experiments were performed using the WEKA[6] machine learning environment. We studied the following learning algorithms in preliminary experiments: RBF SVM, logistic regression, C4.5, and random forest. As the RBF SVM produced the best performance among the different methods, we used this method in our classification experiments, with a nested grid search procedure for parameter tuning, where internal cross-validation is used to find the C and σ parameters of the RBF SVM.

The evaluation was done using stratified 10 times 10-fold-cross-validation and different subsets of attributes are compared. All the methods are compared with the baseline of using the last value of PMI-SO, based on the corrected resampled paired t-student test with an α level of 0.05 (Nadeau and Bengio, 2003). We used the following subsets of attributes: 1) *PMI-SO*: Includes only the feature *pmi-so.last*. This is the baseline and is equivalent to the standard PMI semantic orientation measure, with the decision boundaries provided by the SVM. 2) *ALL*: Includes all the features. 3) *SGD-SO.TS+POS*: Includes all the features from the SGD-SO time series and the POS tag. 4) *PMI-SO.TS+POS*: Includes all the features from the PMI-SO time series and the POS tag. 5) *PMI-SO+POS*: Includes the feature *pmi-so.last* and the POS tag.

We use two evaluation measures that are appropriate for imbalanced datasets:

[6] http://www.cs.waikato.ac.nz/ml/weka/

the weighted area under the ROC curves (AUCs) and the kappa statistic. ROC curves are insensitive to class balance because they include all true positive and false positive rates that are observed (Fawcett, 2006). The kappa statistic is also insensitive to class imbalance because it normalises the classification accuracy by the imbalance of the classes in the data.

The classification results for the four different subsets of attributes in the two datasets are presented in Table 3.8. The symbols + and − correspond to statistically significant improvements and degradations with respect to the baseline, respectively.

We can observe a much lower performance in Neutrality detection than in PosNeg. This indicates that the detection of neutral Twitter words is much harder than distinguishing between positive and negative words. The performance on both datasets tends to be similar. However, the results for STS are better than for ED.EM. This suggests that a balanced collection of positively and negatively labelled tweets is more suitable for lexicon expansion. Another result is that the combination of all features leads to a significant improvement over the baseline for Neutrality and PosNegNeu classification. In the PosNeg classification task, we can see that the baseline is very strong. This suggests that PMI-SO is very good for discriminating between positive and negative words, but not strong enough when neutral words are included. Regarding PMI-SO and SGD-SO time series, we can conclude that they are competitive for Neutrality detection. However, PMI-SO-based features are better for the PosNeg and PosNegNeu tasks.

AUC					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.EM-Neutrality	0.62 ± 0.02	**0.65** ± 0.02 +	**0.65** ± 0.02 +	**0.65** ± 0.02 +	0.64 ± 0.02 +
ED.EM-PosNeg	0.74 ± 0.03	**0.75** ± 0.03	0.71 ± 0.03 -	0.74 ± 0.03	0.73 ± 0.03
ED.EM-PosNegNeu	0.62 ± 0.02	**0.65** ± 0.02 +	0.64 ± 0.02	**0.65** ± 0.02 +	0.64 ± 0.02 +
STS-Neutrality	0.63 ± 0.02	**0.67** ± 0.02 +	0.66 ± 0.02 +	0.66 ± 0.02 +	0.66 ± 0.02 +
STS-PosNeg	**0.77** ± 0.03	**0.77** ± 0.03	0.75 ± 0.03 -	**0.77** ± 0.03	**0.77** ± 0.03
STS-PosNegNeu	0.64 ± 0.02	**0.66** ± 0.01 +	0.65 ± 0.02 +	**0.66** ± 0.02 +	**0.66** ± 0.02 +
Kappa					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.EM-Neutrality	0.23 ± 0.04	**0.3** ± 0.04 +	0.29 ± 0.05 +	**0.3** ± 0.04 +	0.28 ± 0.04 +
ED.EM-PosNeg	0.48 ± 0.06	**0.5** ± 0.06	0.44 ± 0.05	0.49 ± 0.06	0.48 ± 0.06
ED.EM-PosNegNeu	0.28 ± 0.04	**0.33** ± 0.04 +	0.3 ± 0.04	**0.33** ± 0.04 +	0.32 ± 0.04 +
STS-Neutrality	0.26 ± 0.04	**0.33** ± 0.04 +	0.31 ± 0.05 +	0.32 ± 0.04 +	0.32 ± 0.04 +
STS-PosNeg	**0.54** ± 0.06	**0.54** ± 0.06	0.51 ± 0.06 -	0.53 ± 0.06	**0.54** ± 0.05
STS-PosNegNeu	0.31 ± 0.04	**0.35** ± 0.03 +	0.34 ± 0.03 +	0.34 ± 0.03 +	0.34 ± 0.03 +

Table 3.8: World-level classification performance with emoticon-based annotation. Best result per row is given in bold.

Word Classification from Model Transfer Annotated Tweets

We also study the classification of words from the data annotated with the model transfer approach. As described in Section 3.1.1, our soft-annotated collection ED.SL is built by taking STS as the source corpus and a sample of 10 million tweets from ED as the target one. We study two different mechanisms for extracting word-level attributes from the soft-annotated collection of tweets. In the first one, we convert the message-level soft labels into hard ones by imposing different thresholds (λ) and we calculate the same attributes used for the emoticon-annotated data. Taking steps of 0.1, we vary the value of λ from 0.6 to 0.9 and obtain four hard-annotated datasets. In the second approach, we calculate the features directly from the soft labels by relying on the squared loss (Equation 3.5) for building the SGD-SO time series and on partial counts (Equation 3.6) for building the PMI-SO time series.

In this way, we obtain four hard-annotated datasets and one soft-annotated dataset. We calculate the corresponding word-level attributes (see Section 3.1.2) for the $10,000$ most frequent POS-disambiguated words from each of the five datasets. As the most frequent words matching the seed lexicon are not necessarily the same among the different datasets, we take the intersection of them in order to make them comparable. We trained RBF SVMs on the different collections over the intersection of the labelled words using two different feature spaces. In the first one, we use all the attributes, and in the second one, we discard the POS attribute, which is the only feature that is independent of the threshold or the message-level label. The 3-class word-level polarity classification accuracies[7] and kappa values obtained by the different RBF SVMs are shown in Table 3.9.

	ALL		NO POS	
Dataset	Accuracy	Kappa	Accuracy	Kappa
ED.T06	62.82 ± 1.78	**0.34** ± 0.03	61.29 ± 2.02	0.31 ± 0.04
ED.T07	62.77 ± 1.78	**0.34** ± 0.03	**61.60** ± 1.98	**0.32** ± 0.04
ED.T08	62.43 ± 1.83	0.33 ± 0.04	61.03 ± 1.83	0.30 ± 0.03
ED.T09	62.46 ± 1.82	0.33 ± 0.03	60.20 ± 1.89	0.29 ± 0.04
Soft Labels	**63.05** ± 1.81	**0.34** ± 0.03	60.92 ± 2.10	0.30 ± 0.04

Table 3.9: Word classification performance using model transfer. Best result per column is given in bold.

[7] We are not considering AUC in this experiment because all datasets exhibit very similar values for this measure.

The results indicate that the different thresholds and the soft labels produce similar results. Indeed, there are no statistically significant differences among them. However, it is worth mentioning that the soft labels produce a better accuracy than the hard ones when all the attributes are included. Regarding the kappa values, we observe that they become more distinguishable when the POS label is discarded. As ED.T07 achieved the best kappa values for both attribute spaces, we select 0.7 as the best value of λ.

Next, we study the performance of the different feature subsets in data obtained with the model transfer approach. We repeat the previous word-level classification experiments conducted on the emoticon-annotated datasets on the soft-annotated collection (ED.SL) and the best hard-annotated collection (ED.T07). The same four different subsets of attributes are compared and we use again the last value of the PMI-SO series as the baseline. The results are exhibited in Table 3.10.

AUC					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.T07-Neutrality	0.62 ± 0.02	**0.65** ± 0.02 +	**0.65** ± 0.02 +	0.64 ± 0.02	0.64 ± 0.02
ED.T07-PosNeg	0.77 ± 0.03	0.76 ± 0.02	0.74 ± 0.03 -	**0.78** ± 0.03	0.77 ± 0.03
ED.T07-PosNegNeu	0.62 ± 0.02	**0.65** ± 0.02 +	0.65 ± 0.02 +	0.64 ± 0.02	0.64 ± 0.01
ED.SL-Neutrality	0.62 ± 0.02	**0.65** ± 0.02 +	**0.65** ± 0.02 +	0.64 ± 0.02 +	0.64 ± 0.02 +
ED.SL-PosNeg	**0.78** ± 0.03	**0.78** ± 0.03	0.74 ± 0.03 -	**0.78** ± 0.03	**0.78** ± 0.03
ED.SL-PosNegNeu	0.63 ± 0.02	**0.65** ± 0.02 +	0.64 ± 0.02	0.64 ± 0.02	0.64 ± 0.02
Kappa					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.T07-Neutrality	0.23 ± 0.03	0.29 ± 0.04 +	**0.31** ± 0.04 +	0.27 ± 0.04	0.28 ± 0.04
ED.T07-PosNeg	**0.56** ± 0.06	0.54 ± 0.05	0.49 ± 0.06 -	**0.56** ± 0.05	**0.56** ± 0.05
ED.T07-PosNegNeu	0.27 ± 0.03	**0.34** ± 0.04 +	0.33 ± 0.03 +	0.31 ± 0.05	0.31 ± 0.03
ED.SL-Neutrality	0.24 ± 0.05	**0.3** ± 0.04 +	0.29 ± 0.04 +	0.28 ± 0.04 +	0.28 ± 0.04 +
ED.SL-PosNeg	**0.56** ± 0.06	**0.57** ± 0.05	0.49 ± 0.05 -	**0.57** ± 0.05	**0.57** ± 0.06
ED.SL-PosNegNeu	0.3 ± 0.04	**0.33** ± 0.04 +	0.31 ± 0.03	0.32 ± 0.04	0.32 ± 0.04

Table 3.10: World-level classification performance using model transfer. Best result per row is given in bold.

Similarly to the results for the emoticon-annotated experiments shown in Table 3.8, the results are better for PosNeg than for Neutrality, and the combination of all the attributes produces a significant improvement over semantic orientation for 3-class PosNegNeu detection. Indeed, the full attribute space is the only representation that outperforms the baseline for both collections in both AUC and kappa evaluation measures. Another difference between these results and the previous ones is observed for the detection of neutrality. Both PMI-SO and SGD-SO achieved very similar results in the previous experiments, but SGD-SO produces better results here.

We can see that the way in which our different features complement each other becomes clearer when they are calculated from tweets annotated with the model transfer approach.

3.2.3 Lexicon expansion

The ultimate goal of the polarity classification of words is to produce a Twitter-oriented opinion lexicon emulating the properties of SentiWordet, i.e., a lexicon of POS-tagged disambiguated entries with their corresponding distribution for positive, negative, and neutral classes. To do this, we fit logistic regression models to the outputs of the support vector machines trained for the *PosNegNeu* problem, using all the attributes. The resulting models are then used to classify the remaining unlabelled words. This process is performed for the STS, ED.EM, ED.T07, and ED.SL datasets.

A sample from the expanded word list produced with the STS collection is given in Table 3.11. We can see that each entry has the following attributes: the word, the POS-tag, the sentiment label that corresponds to the class with maximum probability, and the distribution. We inspected the expanded lexicon and observed that the estimated probabilities are intuitively plausible. However, there are some words for which the estimated distribution is questionable, such as the word *same* in Table 3.11. We can also observe that words such as *close* and *laugh*, which have more than one POS-tag, receive disambiguated sentiment distributions. We observe that these disambiguations are intuitively plausible as well.

word	POS	label	negative	neutral	positive
alrighty	interjection	positive	0.021	0.087	0.892
anniversary	common.noun	neutral	0.074	0.586	0.339
boooooo	interjection	negative	0.984	0.013	0.003
close	adjective	positive	0.352	0.267	0.381
close	verb	neutral	0.353	0.511	0.136
french	adjective	neutral	0.357	0.358	0.285
handsome	adjective	positive	0.007	0.026	0.968
laugh	common.noun	neutral	0.09	0.504	0.406
laugh	verb	positive	0.057	0.214	0.729
lmaoo	interjection	positive	0.19	0.338	0.472
relaxing	verb	positive	0.064	0.244	0.692
saddest	adjective	negative	0.998	0.002	0
same	adjective	negative	0.604	0.195	0.201
tear	common.noun	negative	0.833	0.124	0.044
wikipedia	proper.noun	neutral	0.102	0.644	0.254

Table 3.11: Example list of words in expanded lexicon.

The provided probabilities can also be used to explore the sentiment inten-

sities of words. In Figure 3.5, we visualise the expanded lexicon intensities of words classified as positive and negative through word clouds. The sizes of the words are proportional to the log odds ratios $\log_2(\frac{P(pos)}{P(neg)})$ and $\log_2(\frac{P(neg)}{P(pos)})$ for positive and negative words, respectively.

(a) (b)

Figure 3.5: Word clouds of positive and negative words using log odds proportions.

3.2.4 Extrinsic Evaluation of the Expanded Lexicons

In this subsection we study the usefulness of our expanded lexicons in an extrinsic task: polarity classification of tweets. This involves categorising entire tweets into a positive or negative sentiment class. The goal of this experiment is to show how the expanded lexicons can be used to improve the message-level classification performance achieved by using the manually annotated seed lexicon and to compare the created resources with two other existing resources that have been widely used for sentiment analysis: SentiWordNet and SentiStrength (Thelwall et al., 2012).

As was previously discussed in Chapter 2, SentiWordNet is a resource in which each WordNet synset is assigned a probability distribution of positive, negative, and neutral classes. Synsets in WordNet are sets of word senses with equivalent meaning. A word with multiple senses or meanings is included in multiple WordNet synsets and, in turn, is associated with multiple sentiment distributions in SentiWordNet. In WordNet, all the senses of a word are ranked according to their frequency of use or popularity. As suggested

in the sample code provided by the SentiWordNet webpage[8], we calculate a weighted average of the sentiment distributions of the synsets in which a POS-disambiguated word occurs in order to obtain a single sentiment distribution for it. The weights correspond to the reciprocal ranks of the senses in order to give higher weights to the most popular senses of a word.

SentiStrength, on the other hand, is a lexicon-based method that returns a positive and negative score for a given tweet.

The evaluation is performed on three collections of tweets that were manually assigned to the positive and negative class. The first collection is *6HumanCoded*[9], which was used to evaluate SentiStrength in the paper that proposed this method. In this dataset, tweets are scored according to positive and negative numerical scores. We use the difference of these scores to create polarity classes and discard messages where it is equal to zero. The other datasets are *Sanders*[10], and *SemEval*[11]. The number of positive and negative tweets per corpus is given in Table 3.12.

	Positive	Negative	Total
6HumanCoded	1340	949	2289
Sanders	570	654	1224
SemEval	5232	2067	7299

Table 3.12: Message-level polarity classification datasets.

We train different logistic regression models on the labelled collections of tweets, based on simple features calculated from the seed lexicon, SentiWordNet, SentiStrength, and from the four expanded lexicons: STS, ED.EM, ED.SL, and ED.T07. For each resource we compute a positive and a negative feature. From the seed lexicon we count the number of positive and negative words matching the content of the tweet. In order to use the POS-disambiguated lexicons such as SentiWordNet and our expanded lexicons, we tag the tweet's words according to POS classes. Then, we calculate the corresponding positive feature by adding the positive probabilities of POS-tagged words labelled as positive within the tweet's content. Likewise, the corresponding negative feature is calculated in an analogous way from the negative probabilities. In the expanded lexicons, words are discarded as non-opinion words whenever

[8] http://sentiwordnet.isti.cnr.it/code/SentiWordNetDemoCode.java
[9] http://sentistrength.wlv.ac.uk/documentation/6humanCodedDataSets.zip
[10] http://www.sananalytics.com/lab/twitter-sentiment/
[11] http://www.cs.york.ac.uk/semeval-2013/task2/

the class with the highest probability corresponds to the neutral one. For SentiStrength we use the positive and negative scores returned by the method for the target tweet.

We study eight different setups based on these attributes. The first one is the seed lexicon baseline which includes only the two attributes calculated from the seed lexicon. The second one corresponds to the SentiWordNet baseline and includes the positive and negative features calculated from it. The third one corresponds to the SentiStrength baseline, which includes the positive and negative scores returned by that method. The next four setups, STS, ED.EM, ED.SL, and ED.707, include the pair of features provided by each corresponding expanded lexicon together with the two features from the seed lexicon. This is done because the expanded lexicons do not contain the words from the seed lexicons that were used to train them. Finally, the last setup, ENS is an ensemble of the four expanded lexicons and the seed lexicon by including the ten features associated with these resources.

In the same way as in the word-level classification task, we use the weighted AUC and the kappa coefficient as evaluation measures, estimated using 10-times 10-fold cross-validation, and we compare the different setups with the three baselines using corrected paired t-tests. The classification results obtained for the different setups are shown in Table 3.13. The statistical significance tests of each setup with respect to each of the three baselines (seed lexicon, SentiWordNet, and SentiStrength) are indicated by a sequence of three symbols. Improvements are denoted by a plus (+), degradations by a minus (-), and cases where no statistical significant difference is observed by an equal sign (=).

The results indicate that the expanded lexicons produce meaningful improvements in performance over the seed lexicon and over SentiWordNet on the different datasets. We believe that the reason why SentiWordNet is not achieving good results is its lack of informal English expressions. SentiStrength, on the other hand, is a strong baseline for Twitter sentiment analysis. This is because of two reasons: 1) it is based on a lexicon formed by both formal and informal English words, and 2) it includes linguistic rules for handling negations and intensifiers. We observe that most of our expanded lexicons are at least competitive with SentiStrength according to the statistical tests. Moreover, there are several cases in Sanders and SemEval in which the expanded lexicons achieve statistically significant improvements over SentiStrength, especially for AUC. This is noteworthy, considering that the fea-

AUC			
Dataset	6HumanCoded	Sanders	SemEval
Seed.Lex	0.77 ± 0.03 = + -	0.77 ± 0.04 = + =	0.77 ± 0.02 = + -
SW	0.74 ± 0.03 - = -	0.7 ± 0.05 - = -	0.76 ± 0.02 = = -
SS	0.81 ± 0.02 + + =	0.78 ± 0.03 = + =	0.81 ± 0.02 + + =
STS	0.82 ± 0.02 + + =	**0.84** ± 0.04 + + +	**0.83** ± 0.02 + + +
ED.EM	0.82 ± 0.03 + + =	0.83 ± 0.04 + + +	0.81 ± 0.02 + + =
ED.SL	0.81 ± 0.02 + + =	0.83 ± 0.04 + + +	0.81 ± 0.02 + + =
ED.T07	0.81 ± 0.03 + + =	0.83 ± 0.04 + + +	0.82 ± 0.02 + + +
ENS	**0.83** ± 0.02 + + =	**0.84** ± 0.04 + + +	*0.83* ± 0.02 + + +
Kappa			
Dataset	6HumanCoded	Sanders	SemEval
Seed Lex	0.4 ± 0.06 = + -	0.42 ± 0.08 = + =	0.35 ± 0.04 = + -
SW	0.32 ± 0.06 - = -	0.26 ± 0.1 - = -	0.3 ± 0.04 - = -
SS	**0.52** ± 0.05 + + =	0.45 ± 0.06 = + =	0.38 ± 0.03 + + =
STS	0.47 ± 0.06 + + =	**0.55** ± 0.08 + + +	0.38 ± 0.04 + + =
ED.EM	0.47 ± 0.05 + + -	0.54 ± 0.07 + + +	0.35 ± 0.04 = + -
ED.SL	0.46 ± 0.05 + + -	0.54 ± 0.08 + + +	0.36 ± 0.04 = + =
ED.T07	0.47 ± 0.05 + + -	0.53 ± 0.08 + + +	0.4 ± 0.04 + + =
ENS	0.49 ± 0.05 + + =	0.54 ± 0.07 + + +	**0.42** ± 0.04 + + +

Table 3.13: Message-level polarity classification performance. Best result per column is given in bold.

tures we calculate from the expanded lexicons are based on simple additions of prior sentiment scores in contrast to the linguistic rules that SentiStrength uses for aggregating its lexicon's words. Most of the cases where the expanded lexicons are statistically significantly worse than SentiStrength occur for the kappa measure in *6HumanCoded*.

Regarding the lexicons built from emoticon-annotated data, the performance of STS is slightly better than that of ED.EM. This pattern was also observed in the word-level classification performance shown in Table 3.8. This suggests that the two different ways of evaluating the lexicon expansion, one at the word level and the other at the message level, are consistent with each other. Regarding the lexicons built from the model transfer annotation approach, the results are competitive with the ones achieved with the emoticon-annotated data. Moreover, the lexicon built from hard transfered labels appears to be slightly better than the one built using soft labels, especially in the kappa value for the *SemEval* dataset.

We can also observe, in the majority of the cases, that the best performance

is obtained by the ensemble of expanded lexicons. Therefore, we can conclude that lexicons expanded using either data from different collections or by applying different annotation approaches can be combined to improve message-level classification of tweets.

As was discussed in the previous section, the seed lexicon does not provide POS-tagged entries. Therefore, words exhibiting multiple POS-tags were labelled with the same polarity in the word-level training data. We also mentioned that this assumption can make the classifier learn spurious patterns and erroneously classify unlabelled words. For example the word *ill* together with the POS tag *nominal+verb* receives a negative label. However, when *ill* is used with the part-of-speech tag that refers to the contraction of the pronoun *I* and the verb *will*, it should be labelled as neutral instead. Moreover, by inspection we realised that *ill* is the only labelled entry with the POS tag *nominal+verb*. Considering that the POS tag is also used as a feature in the word-level classifiers, we observed that most of the words exhibiting the POS tag *nominal+verb* were classified into the negative class in all expanded lexicons. Common sense suggests that these words should be expanded to the neutral class.

In order to avoid learning this type of spurious pattern, we re-trained the word-level classifiers using an outlier removal technique. More specifically, we clean out the instances from the word-level training data that are misclassified by a classifier evaluated using 10-fold cross-validation on this data, again using an RBF SVM. Afterwards, we retrain the word-level classifiers on the cleaned data and create new versions of all the expanded lexicons. We inspected the new versions of the expanded lexicons observing that words exhibiting the *nominal+verb* POS tag are classified to the neutral class as common sense suggests. This shows that removing outliers is a successful way of tackling ambiguities such as the one produced by the word *ill* in the seed lexicon.

The message-level classification results obtained by the expanded lexicons with outlier removal are shown in Table 3.14. The improvements or degradations over the previous expanded lexicons are denoted with symbols $\uparrow$ and $\downarrow$ respectively. In relation to *6HumanCoded*, we observe that the AUC metric is improved for almost all the different setups, and that all the setups outperform the previous kappa results. It is noteworthy that the kappa value achieved by the ensemble of lexicons in this dataset exceeds the previous value by 0.03. On the other hand, we see degradations in the kappa values for *Sanders*. Regarding *SemEval*, we see a degradation with STS, a substantial improvement

AUC			
Dataset	6HumanCoded	Sanders	SemEval
STS	0.82 ± 0.03 + + =	0.82 ± 0.04 + + + ↓	0.82 ± 0.02 + + + ↓
ED.EM	0.84 ± 0.02 + + + ↑	0.83 ± 0.04 + + +	0.83 ± 0.02 + + + ↑
ED.SL	0.82 ± 0.03 + + = ↑	0.83 ± 0.04 + + +	0.82 ± 0.02 + + = ↑
ED.T07	0.81 ± 0.03 + + =	0.83 ± 0.04 + + +	0.82 ± 0.02 + + +
ENS	**0.84** ± 0.02 + + + ↑	**0.84** ± 0.04 + + +	**0.84** ± 0.02 + + + ↑
Kappa			
Dataset	6HumanCoded	Sanders	SemEval
STS	0.48 ± 0.05 + + = ↑	0.52 ± 0.08 + + = ↓	0.36 ± 0.03 = + = ↓
ED.EM	0.51 ± 0.05 + + = ↑	**0.53** ± 0.08 + + + ↓	0.41 ± 0.03 + + = ↑
ED.SL	0.48 ± 0.05 + + = ↑	**0.53** ± 0.08 + + + ↓	0.37 ± 0.04 = + = ↑
ED.T07	0.48 ± 0.06 + + = ↑	**0.53** ± 0.08 + + +	0.39 ± 0.04 + + = ↓
ENS	**0.52** ± 0.05 + + = ↑	**0.53** ± 0.07 + + + ↓	**0.42** ± 0.03 + + +

Table 3.14: Message-level polarity classification performance with outlier removal. Best result per columns is given in bold.

with ED.EM, and a minor improvement with ED.SL. Another interesting result is that the removal of outliers creates lexicons that are always equal or better than SentiStrength according to the statistical tests.

The fact that the removal of outliers can also produce degradations in the quality of the expanded lexicons, as in the case of the *Sanders* dataset, indicates that words that are useful for learning the word-level classifier have been removed. This suggests that the problem of reducing the noise in the seed lexicon is hard to address in an automatic fashion. A simple but labour-intensive approach to overcome this problem would be to manually clean the labelled POS-disambiguated words.

3.3 Discussion

In this chapter, we have presented a method for opinion lexicon expansion in the context of tweets. The method exploits information from three types of information sources, all of which are relatively cheap to obtain: emoticon-annotated tweets, unlabelled tweets, and hand-annotated lexicons. The method creates a lexical resource with disambiguated POS entries and a probability distribution for positive, negative, and neutral classes. To the best of our knowledge, our method is the first approach for creating a Twitter opinion lexicon with these characteristics. Considering that these characteristics are very similar to those of SentiWordNet, a well-known publicly available lexical

resource, we believe that several sentiment analysis methods that are based on SentiWordNet can be easily adapted to Twitter by relying on our expanded lexicons[12]. Moreover, our expanded resources have shown to outperform the tweet-level polarity classification performance achieved by SentiWordNet and SentiStrength in most cases.

The word-level experimental results show that the supervised fusion of POS tags, SGD-SO, and PMI-SO, produces a significant improvement for three-dimensional word-level polarity classification compared to using PMI semantic orientation alone. We can also conclude that attributes describing the central location of SGD-SO and PMI-SO time series tend to be more informative than the last values of the series because they smooth the temporal fluctuations in the sentiment pattern of a word.

There are many domains, such as politics, in which emoticons are not frequently used to express positive and negative opinions. This is an important limitation of previous approaches for domain-specific lexicon expansion that are based solely on emoticon-annotated tweets. The proposed model transfer annotation approach tackles this problem and enables inference of opinion words from any collection of unlabelled tweets.

We have also proposed a novel way for computing word-level attributes from data with soft labels. The proposed soft version of PMI-SO based on partial counts can be used for expanding lexicons from any collection of tweets in an unsupervised fashion. In contrast to a threshold approach, soft PMI-SO is parameter free and avoids discarding tweets that may contain valuable words.

[12]The expanded lexicons and the source code used to generate them are available for download at http://www.cs.waikato.ac.nz/ml/sa/lex.html#kbs16.

Chapter 4

Distributional Models for Affective Lexicon Induction

The Distributional Hypothesis (Harris, 1954) states that words occurring in similar contexts tend to have similar meanings. This hypothesis is exploited in this chapter for inducing polarity and affective words from Twitter corpora. To this end, words are represented by distributional vectors. Distributional vectors (Turney and Pantel, 2010) are used for representing lexical items such as words according to the context in which they occur in a corpus of documents or tweets. In other words, distributional models infer the meaning of a word from the distribution of the words that surround it.

The main benefit of distributional models over the approach presented in Chapter 3, is that distributional models do not depend on tweets labelled by sentiment. We experiment with two distributional approaches:

1. The tweet centroid model, which creates word-vectors from tweet-level attributes (e.g., unigrams and Brown clusters) by averaging all the tweets in which the target word appears.

2. Word embeddings (Mikolov et al., 2013), which are low-dimensional continuous dense word vectors trained from document corpora.

This chapter is divided into two major parts: 1) a first part focused on classifying words into positive, negative, and neutral polarity classes using the tweet centroid model, and 2) an extended study aimed at producing a more fine-grained word-level categorisation based on multi-label emotion categories, such as anger, fear, surprise, and joy. We explore different types of word-level features derived from tweet centroids and word embeddings using multi-label classification techniques. We close the chapter with a discussion of the main findings of the study.

4.1 Polarity Lexicon Induction with Tweet Centroids

The rationale of using distributional models for polarity lexicon induction is that words exhibiting a certain polarity are more likely to be used in contexts expressing the same polarity than in contexts exhibiting a different one. Thus, the context of a word can determine its polarity. The tweet centroid model we propose in this chapter is a distributional representation that exploits the short nature of tweets by treating them as the whole contexts of words. This is done by representing words as the centroids of the tweets in which they occur within a corpus of tweets.

Suppose we have a corpus $\mathcal{C}$ formed by n tweets $t_1, \ldots, t_n$, where each tweet t is a sequence of words. Let $\mathcal{V}$ be the vocabulary formed by the m different words $w_1, \ldots, w_m$ found in $\mathcal{C}$. The tweets from $\mathcal{C}$ are represented by feature vectors x of dimensionality k. Note that different NLP features can be used to form the tweet vectors. A standard approach is to use the vector space model or bag-of-words model introduced in Chapter 1.

For each word w, we define the word-tweet set $\mathcal{M}(w)$ as the set of tweets in which w is observed:

$$\mathcal{M}(w) = \{m : w \in m\} \tag{4.1}$$

We define the tweet centroid word vector $\overrightarrow{w}$ as the centroid of all tweet vectors in which w is used. In other words, $\overrightarrow{w}$ is a k-dimensional vector in which each dimension w_j is calculated as follows:

$$w_j = \sum_{t \in \mathcal{M}(w)} \frac{x_j^{(t)}}{|\mathcal{M}(w)|} \tag{4.2}$$

Another interpretation of the tweet centroid model is that words are treated as the expected tweet in which they might occur.

In this study, we build the word vectors from two different tweet-level representation. The first is a high-dimensional bag-of-words model using unigram frequencies as dimension values. The second is a semantic representation based on word clusters. We employ the Brown clustering algorithm (Brown et al., 1992) to tag a tweet with a sequence of word clusters and create cluster frequency vectors. These models are formalised as follows.

The tweet-level unigram model represents each tweet t as an m-dimensional vector $\overrightarrow{tb}$ where each dimension j has a numerical value $f_j(t)$ that corresponds to the frequency of the word w_j within the sequence of words in t. We define the unigram word-level vector $\overrightarrow{wb}$ as as the centroid of all tweet vectors in

which w is used.

However, because unigram models tend to produce high-dimensional sparse vectors, we also study another word vector representation with lower dimensionality that is based on the interaction of word clusters.

Let c be a clustering function that maps the m words from $\mathcal{V}$ to a partition $\mathcal{S}$ containing l classes, with $l \ll m$. In our experiments, this function is trained in an unsupervised fashion from a corpus of tweets using the Brown clustering algorithm (Brown et al., 1992), which produces hierarchical clusters by maximising the mutual information of bigrams. These clusters have shown to be useful for tagging tweets according to part-of-speech classes (Gimpel et al., 2011).

We tag the word sequences of the tweets from $\mathcal{C}$ with the clustering function c. Afterwards, we create a new tweet-level vector $\overrightarrow{tc}$ of l dimensions based on the frequency of occurrence of a cluster s in the tweet. The cluster-based word vectors $\overrightarrow{wc}$ are calculated analogously to the bag-of-words vectors in the first approach. We take the centroids of the cluster-based vectors $\overrightarrow{tc}$ from the tweets of $\mathcal{W}(w)$, producing l-dimensional vectors for each word.

We will classify each word from a corpus of unlabelled tweets into one of three different polarity classes: positive, negative, or neutral. As in the previous chapter, we use a seed lexicon to label a sample of the words and train a classifier on the labelled instances represented by tweet centroids. The fitted model is then used to classify the remaining unlabelled words. A diagram of how the tweet centroid vectors are used for lexicon induction using supervised learning is shown in Figure 4.1.

4.1.1 Tweet Centroids and Word-Context Matrices

The tweet centroid model can be viewed as a variation of the word-context matrix[1] for semantic modelling (Turney and Pantel, 2010). This matrix has dimensionality $|\mathcal{V}| \times |\mathcal{V}|$, and each cell (i, j) is a co-occurrence based association value between a target word w_i and a context word w_j calculated from a corpus of documents. Contexts can be represented by entire documents. However, considering that common documents (e.g., Wikipedia articles, news articles, scientific papers) are formed by multiple sentences and discuss different ideas, it is normally preferred to use windows of words surrounding the target one. The window length is a user-specified parameter that is usually between 1 and 8 words on both the left and the right sides of the target word, i.e., the total

[1] This matrix is also named word-word or term-term matrix.

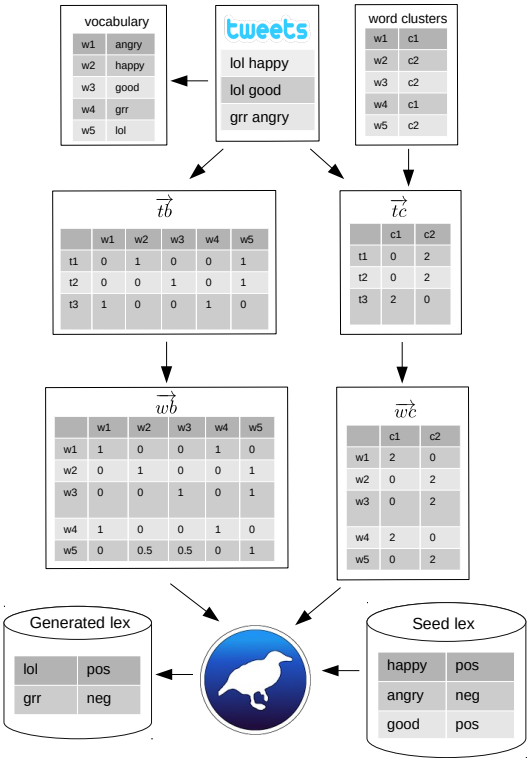

Figure 4.1: Twitter-lexicon induction with tweet centroids. The bird represents the Weka machine learning software.

contexts are usually between 3 and 17 words. Whereas shorter windows are likely to capture syntactic information, longer windows are more useful for representing meaning (Jurafsky and Martin, 2008). The associations between words can be calculated using different approaches such as: co-occurrence counts, positive point-wise mutual information (PPMI), and the significance values of a paired t-test. The most common of those according to (Jurafsky and Martin, 2008) is PPMI. This measure is a filtered version of the traditional PMI measure in which negative values are set to zero:

$$\text{PPMI}(w_i, w_j) = \max(0, \text{PMI}(w_i, w_j)) \tag{4.3}$$

PMI alone calculates the log of the probability of both words occurring together over the probability of both words being independent. Negative PMI values suggest that the two words co-occur less often than chance. These estimates are unreliable unless the counts are calculated from very large corpora (Jurafsky and Martin, 2008). PPMI corrects this problem by replacing negative values by zero.

The tweet centroid model calculated with unigrams is analogous to a word-

context matrix in which whole tweets are used as context and where the word-word associations are calculated using co-occurrence counts normalised by the numbers of contexts where the target word appears. Considering that the average number of words in a tweet is around 14 (Guo, Li, Ji and Diab, 2013), using tweets as contexts is similar to using windows of 14 words when building semantic vectors from longer documents.

A noteworthy property of the tweet centroid model is that it admits arbitrary representations of contexts such as Brown clusters or other features capable of capturing useful semantic information. Traditional word-context matrices lack this property.

4.1.2 Evaluation

The evaluation of the tweet centroids model for lexicon induction is performed in two parts: 1) an intrinsic evaluation, and 2) an extrinsic evaluation.

Intrinsic Evaluation

The tweets we use in our experiments are taken from the same collections used in Chapter 3. We take a random sample of 2.5 million English tweets from the Edinburgh corpus (ED) (Petrović et al., 2010), and we use the STS emoticon-annotated corpus with the emoticons removed from the content[2]. The ED corpus represents a realistic sample from a stream of tweets, whereas STS was intentionally manipulated to over-represent subjective tweets.

We tokenise the tweets from both collections and create the vectors $\overrightarrow{wb}$ and $\overrightarrow{wc}$ as described in Section 4.1. The clustering function c was taken from the **TweetNLP** project[3]. This function was trained to produce 1000 different word clusters from a collection of around 56 million tweets using the Brown clustering algorithm.

The two vectors $\overrightarrow{wb}, \overrightarrow{wc}$ are used as attribute vectors to train a word-level classifier for lexicon induction. To avoid learning spurious relationships from infrequent words, vectors of words that occur in fewer than 10 tweets are discarded ($|\mathcal{W}(w)| < 10$). We also discard the dimensions from $\overrightarrow{wb}$ that correspond to those infrequent words. Analogously, we remove all dimensions from $\overrightarrow{wc}$ that are associated with clusters appearing in fewer than 10 tweets.

We label the words that match a seed lexicon into three sentiment categories: positive, negative, and neutral. The seed lexicon is the same as the

[2]Emoticons are not used as tweet-level sentiment labels in the current approach.
[3]http://www.ark.cs.cmu.edu/TweetNLP/

one used in Chapter 3 after including a list of 87 positive and negative emoticons. Note that emoticons are only present in the ED corpus.

The seed lexicon, after including the emoticon list, has 3769 positive, 6414 negative, and 7088 neutral words. The main properties of the ED and STS datasets are summarised in Table 4.1.

Dataset	STS	ED
#tweets	$1,600,000$	$2,500,000$
#positive words	$2,015$	$2,639$
#negative words	$2,621$	$3,642$
#neutral words	$3,935$	$5,085$
#unlabelled words	$36,451$	$67,692$
#unigram attributes	$45,022$	$79,058$
#cluster-vector attributes	993	999

Table 4.1: Dataset properties.

We first study the problem of classifying words into positive and negative classes. We train an L2-regularised logistic regression model with the regularisation C parameter set to 1.0 using LibLINEAR. For performance estimation, we apply 10 times 10-folds cross-validation on the positive and negative labelled words from the two datasets. We compare three different instance spaces: unigram vectors $\overrightarrow{wb}$, Brown cluster vectors $\overrightarrow{wc}$, and the concatenation of both: $[wb_1, \ldots, wb_m, wc_1, \ldots, wc_k]$. We compare classification accuracy and the weighted area under the ROC curve (AUC) obtained by the different instance spaces using a corrected resampled paired t-student test with an α level of 0.05. Results are displayed in Table 4.2. Statistically significant improvements over the bag-of-words approach are denoted with the symbol $+$.

Accuracy			
Dataset	UNI	CLUSTER	CONCAT
STS	75.52 ± 1.81	77.2 ± 1.9 +	**77.85** ± 1.94 +
ED	77.75 ± 1.54	77.62 ± 1.37	**79.15** ± 1.39 +
AUC			
Dataset	UNI	CLUSTER	CONCAT
STS	0.83 ± 0.02	0.84 ± 0.02 +	**0.85** ± 0.02 +
ED	0.85 ± 0.01	0.85 ± 0.01	**0.86** ± 0.01 +

Table 4.2: Word-level 2-class polarity classification performance.

We can observe that the classification results are slightly better for ED than for STS. The cluster-based representation is better than the unigram representation in STS. However, this pattern is not observed in ED. The concatenation of both vector models produces significant improvements in accuracy and AUC over the baseline in both datasets.

Accuracy			
Dataset	UNI	CLUSTER	CONCAT
STS	61.84 ± 1.46	64.42 ± 1.54 +	**64.57** ± 1.44 +
ED	62.93 ± 1.31	64.5 ± 1.16 +	**65.5** ± 1.19 +
AUC			
Dataset	UNI	CLUSTER	CONCAT
STS	0.77 ± 0.01	**0.79** ± 0.01 +	**0.79** ± 0.01 +
ED	0.78 ± 0.01	0.79 ± 0.01 +	**0.8** ± 0.01 +

Table 4.3: Word-level three-class polarity classification performance.

In the next experiment, we include neutral words to train a 3-class classifier capable of classifying words as positive, negative, or neutral. The classification results are given in Table 4.3. We can see that the classification performance is lower than in the previous experiment. The cluster-based vectors are significantly better than the unigram vectors in both datasets. This suggests that word clusters are especially helpful in distinguishing neutral and non-neutral words. The concatenation of the two vectors achieves the best performance among all the experiments.

We also conduct another intrinsic evaluation in which we compare the tweet centroid model based on unigram features with a widely used distributional representation for word semantics: positive point-wise mutual information (PPMI). Additionally, we study how the size of the input corpus from which the word vectors are drawn, affects the polarity classification of words.

The PPMI word vectors are taken from the rows of a word-context matrix built from a corpus of tweets, in which tweets are used as contexts. Each cell in the matrix corresponds to the PPMI between the target word w_i and the context word w_j:

$$\text{PPMI}(w_i, w_j) = \max\left(0, \log_2\left(\frac{Pr(w_i \wedge w_j)}{Pr(w_i)Pr(w_j)}\right)\right). \tag{4.4}$$

Let $\text{count}(w_i)$ be the number of contexts (tweets in our case) where w_i is observed in a corpus of n tweets, and $\text{count}(w_i, w_j)$ be the number of tweets where words w_i and w_j co-occur. The probability of observing a word w_i is estimated as follows:

$$Pr(w_i) = \frac{\text{count}(w_i)}{n}, \tag{4.5}$$

and the joint probability of words w_i and w_j is calculated as:

$$Pr(w_i \wedge w_j) = \frac{\text{count}(w_i, w_j)}{n}. \tag{4.6}$$

Hence, the PPMI score between two words is:

$$\text{PPMI}(w_i, w_j) = \max\left(0, \log_2\left(\frac{\text{count}(w_i, w_j) \times n}{\text{count}(w_i) \times \text{count}(w_j)}\right)\right). \quad (4.7)$$

We use Laplace smoothing to avoid zero counts. We train L_2-regularised logistic regression models on word vectors represented with tweet centroids (using only unigrams) and PPMI. The word vectors are calculated from samples of the Edinburgh corpus ranging from $100,000$ to $3,200,000$ tweets. Infrequent words are discarded in the same way as in the previous experiment. We study the performance of 2-class and 3-class word-level polarity classification with 10-fold cross-validation according to three evaluation measures: accuracy, AUC, and kappa. Additional variables we report are: the processing time (in minutes) for building the word vectors, the number of attributes of the word vectors, the average fraction of active attributes (attributes with non-zero values), and the number of training instances for each classification task. The results are shown in Table 4.4.

Tweet Centroids (UNI)						
Tweets	$100,000$	$200,000$	$400,000$	$800,000$	$1,600,000$	$3,200,000$
Processing Time (Minutes)	0.463	0.965	1.973	4.167	8.302	16.373
Number of attributes	9077	14335	22132	35051	57838	100002
Fraction of active attributes	4.7%	3.9%	3.3%	2.8%	2.3%	1.8%
Train instances 2class	1801	2557	3455	4531	5658	6685
Accuracy 2class	73.792	75.675	74.877	76.650	78.226	78.235
Weighted AUC 2class	0.802	0.825	0.822	0.837	0.851	0.853
Kappa 2class	0.462	0.512	0.497	0.527	0.550	0.540
Train instances 3class	3541	4955	6550	8410	10334	12083
Accuracy 3class	57.583	60.686	60.351	61.332	62.735	63.436
Weighted AUC 3class	0.712	0.744	0.746	0.765	0.779	0.790
Kappa 3class	0.267	0.333	0.338	0.364	0.392	0.405
PPMI						
Processing Time (Minutes)	0.472	0.975	1.958	4.107	8.132	16.764
Number of attributes	9077	14335	22132	35051	57838	100002
Fraction of active attributes	4.7%	3.9%	3.3%	2.8%	2.4%	1.8%
Train Instances 2class	1801	2557	3455	4531	5658	6685
Accuracy 2class	71.294	73.993	72.243	74.487	74.850	77.188
Weighted AUC 2class	0.790	0.808	0.800	0.815	0.820	0.835
Kappa 2class	0.418	0.479	0.443	0.484	0.483	0.522
Train Instances 3class	3541	4955	6550	8410	10334	12083
Accuracy 3class	51.624	52.654	53.420	54.792	54.838	57.039
Weighted AUC 3class	0.646	0.666	0.686	0.695	0.700	0.719
Kappa 3class	0.213	0.240	0.259	0.288	0.291	0.324

Table 4.4: Intrinsic Evaluation of tweet centroids and PPMI for lexicon induction.

From the table we can observe that the tweet centroids produce better distributional vectors than PPMI for 2-class and 3-class polarity classification according to the three performance measures. We also observe that the size of the corpus has a positive impact on the classification performance in the

two types of word vectors. The processing time and the dimensionality of the vectors increase when increasing the size of input corpus. The same is true for words that match the seed lexicon used for training the classifiers. The word vectors are very sparse in both distributional approaches, i.e., the average number of dimensions different from zero is less than the 5% of the total number of attributes in all cases.

Extrinsic Evaluation

We use the three-class world-level classifiers trained using the concatenation of the unigrams and word clusters to label the unlabelled words from STS and from the 2.5 million instance sample of ED. A sample of the induced words from the ED sample with the estimated probabilities for negative, neutral, and positive classes is shown in Table 4.5.

word	label	negative	neutral	positive
#recession	negative	0.603	0.355	0.042
#silicon_valley	neutral	0.043	0.609	0.348
bestfriends	positive	0.225	0.298	0.477
christamas	positive	0.003	0.245	0.751
comercials	negative	0.678	0.317	0.005
hhahaha	positive	0.112	0.409	0.479
powerpoint	neutral	0.068	0.802	0.13
psychotic	negative	0.838	0.138	0.024
widows	negative	0.464	0.261	0.275
yassss	positive	0.396	0.08	0.524

Table 4.5: Example of induced words.

As an additional validation for the induced words, we study their usefulness for classifying the overall polarity of Twitter messages. To do this, we compare the classification performance obtained by a simple classifier that uses attributes calculated from the seed lexicon, with the performance obtained by a classifier with attributes derived from both the seed lexicon and the induced words. This evaluation is analogous to the extrinsic evaluation conducted in Chapter 3 and is done on the same three collections of tweets: 1) *6Human-Coded*, 2) *Sanders*, and 3) *SemEval*.

The baseline of this experiment is a logistic regression model trained using the number of positive and negative words from the seed lexicon that are found within the tweet's content as attributes. For each expanded lexicon, we train a logistic regression model using the baseline attributes together with a positive and a negative score calculated as the sum of the corresponding probabilities of words classified as positive or negative, respectively.

Accuracy			
Dataset	Baseline	STS	ED
Sanders	73.25 ± 3.51	74.76 ± 4.21	**76.58** ± 3.8 +
6HumanCoded	72.84 ± 2.57	75.08 ± 2.31 +	**76.42** ± 2.34 +
SemEval	77.72 ± 1.24	78.97 ± 1.31 +	**79.18** ± 1.22 +
AUC			
Dataset	Baseline	STS	ED
Sanders	0.78 ± 0.04	0.8 ± 0.04 +	**0.83** ± 0.04 +
6HumanCoded	0.79 ± 0.03	0.82 ± 0.03 +	**0.83** ± 0.02 +
SemEval	0.78 ± 0.02	0.82 ± 0.02 +	**0.84** ± 0.02 +

Table 4.6: Message-level classification performance.

The classification results obtained for message-level classification in the three datasets are shown in Table 4.6. We observe from the table that with the exception of the accuracy obtained by the STS-based lexicon on the Sanders dataset, the induced lexicons produce significant improvements over the baseline. Furthermore, the lexicon induced from the ED corpus outperforms the STS lexicon in accuracy and AUC score respectively. These results indicate that collections of tweets manipulated to over-represent subjective tweets, such as STS, are not necessarily better for lexicon induction than random collections of tweets such as ED.

The AUC results obtained with the lexicon induced from ED are very similar to the results obtained by the lexicons induced with the word-sentiment associations from Chapter 3 (Table 3.13). It is interesting to observe that lexicons built using word representations of different nature perform very similarly when evaluated on the same task.

We also conduct an extrinsic evaluation of the lexicons built in the second part of our intrinsic evaluation. The lexicons are built using classifiers trained on tweet centroids using unigrams and on word vectors calculated with PPMI. The word vectors are calculated from the same samples of ED as used in Table 4.4. We report accuracy, AUC, and kappa, of logistic regression models trained on labelled tweets from SemEval, Sanders, and 6HumanCoded. We use the same lexicon-based attributes used in the previous experiment. The results are given in Table 4.7.

We observe from the table that word centroids produce better lexicons than PPMI in the extrinsic evaluation task. We observe mixed results in relation to the size of the input corpus. While lexicons built from larger corpora perform better in Sanders and 6HumanCoded, they perform worse in SemEval. We believe that increasing the input corpus size can produce lexicons with noisy information as discussed below.

Tweet Centroids (UNI)						
Tweets	$100,000$	$200,000$	$400,000$	$800,000$	$1,600,000$	$3,200,000$
Accuracy SemEval	79.066	79.066	78.888	78.586	78.709	78.764
Weighted AUC SemEval	0.827	0.825	0.828	0.830	0.825	0.827
Kappa SemEval	0.430	0.428	0.425	0.418	0.419	0.422
Accuracy Sanders	75.980	75.408	76.389	76.307	76.797	77.124
AUC Sanders	0.822	0.825	0.825	0.825	0.825	0.830
Kappa Sanders	0.518	0.507	0.526	0.524	0.534	0.540
Accuracy 6HumanCoded	75.186	74.705	74.924	74.487	75.273	75.841
AUC 6HumanCoded	0.822	0.822	0.823	0.818	0.824	0.828
Kappa 6HumanCoded	0.477	0.466	0.473	0.462	0.481	0.494
PPMI						
Accuracy SemEval	77.161	77.257	77.339	77.394	77.463	77.504
Weighted AUC SemEval	0.790	0.787	0.786	0.785	0.788	0.787
Kappa SemEval	0.359	0.361	0.365	0.367	0.367	0.368
Accuracy Sanders	73.366	73.611	74.592	74.020	73.856	74.101
AUC Sanders	0.794	0.800	0.802	0.799	0.804	0.808
Kappa Sanders	0.466	0.470	0.490	0.478	0.475	0.479
Accuracy 6HumanCoded	73.526	72.608	73.526	72.870	72.783	73.569
AUC 6HumanCoded	0.800	0.798	0.798	0.793	0.795	0.799
Kappa 6HumanCoded	0.439	0.420	0.440	0.425	0.424	0.440

Table 4.7: Extrinsic Evaluation of TCM and PPMI for lexicon induction.

Suppose we have two lexicons $\mathcal{L}_s$ and $\mathcal{L}_b$, built from the words occurring in two collections of unlabelled tweets $\mathcal{C}_s$ and $\mathcal{C}_b$, respectively, and where $\mathcal{C}_b$ is much larger than $\mathcal{C}_s$ ($|\mathcal{C}_b| \gg |\mathcal{C}_s|$). According to the heaps law (Manning et al., 2008), the number of different words in a corpus increases in a logarithmic fashion with the size of the corpus. Hence, $\mathcal{L}_b > \mathcal{L}_s$.

The vectors of the words that are included in both lexicons $\mathcal{L}_s \cap \mathcal{L}_b$ are more likely to have been built from more context in $\mathcal{L}_b$ than in $\mathcal{L}_s$. Thus, these words are probably better represented in $\mathcal{L}_b$. However, the new words that are only included in $\mathcal{L}_b$ will not necessarily contain enough context for being accurately classified by polarity. This suggests that lexicons built from large collections of tweets are prone to include many words with noisy information.

4.2 Inducing Word–Emotion Associations by Multi-label Classification

Analysing the emotions expressed in Twitter has important applications in the study of public opinion. Word-emotion association lexicons, which are lists of terms annotated according to emotional categories, are widely used resources for analysing emotions in textual passages. The NRC word-emotion association lexicon (NRC-10)[4] (Mohammad and Turney, 2013) is a well-known lexical resource for emotion analysis created by crowdsourcing via Mechanical Turk.

[4] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

It contains 14,182 distinct English words manually annotated according to ten non-exclusive binary categories including the eight emotions from Plutchik's wheel of emotions (Plutchik, 2001): joy, sadness, anger, surprise, fear, disgust, trust and anticipation; and two sentiment classes: positive and negative. For example, the word **achieved** is mapped into the categories anticipation, joy, trust, and positive, and the word **exile** is mapped into anger, fear, sadness, and negative. There are 7,714 words that are not associated with any affective category and can be considered neutral, such as powder and corn. NRC-10 does not cover informal expressions commonly used in social media such as hashtags, slang words and misspelled words, and consequently suffers from limitations when analysing emotions from microblogging messages such as tweets.

In this section, we study how to automatically expand NRC-10 with the words found in a corpus of unlabelled tweets. The expansion is performed using multi-label classification techniques. These techniques assign instances to multiple non-exclusive classes such as the ones provided by NRC-10. We represent words using different types of features derived from our proposed tweet centroid model and low-dimensional embeddings.

The words from NRC-10 that occur in the corpus are labelled according to the emotional categories provided by the lexicon. The feature vectors for the words along with these affect labels are used for learning a word-level multi-label affect classifier. As some categories from NRC-10 correlate with each other, we explore multi-label classification techniques that exploit label co-occurrence such as classifier chains (Read, Pfahringer, Holmes and Frank, 2011). The fitted multi-label classification model is then used to classify the remaining unlabelled words into emotions.

To the best of our knowledge, this is the first emotion lexicon expansion model for tweets in which a word-level multi-label classifier is trained using features calculated from unlabelled corpora.

Probably the most related work to ours, which was already described in Chapter 2, was proposed in (Mohammad and Kiritchenko, 2015). In that work, an emotion-oriented lexicon is built using PMI associations from tweets annotated with emotion-oriented hashtags: #anger, #disgust, #fear, #happy, #sadness, and #surprise. There are two limitations of this approach: 1) words that do not co-occur with those emotion-oriented hashtags will be excluded, and 2) there are many domains in which hashtags are not frequently used to express emotions, and for which this approach would be unsuitable for creating

domain-specific emotion lexicons. In contrast, our approach takes a target corpus of unlabelled tweets from any domain and a seed lexicon to perform the induction.

4.2.1 Multi-label Classification of Words into Emotions

Analogously to the process conducted in Section 4.1, the first step is to tokenise and extract word-level features from a target corpus of ten million unlabelled tweets written in English taken from the Edinburgh corpus (ED). We use two models for extracting word-level features: 1) the tweet centroid model, and 2) the skip-gram model (Mikolov et al., 2013). For the tweet centroid model we again consider unigrams (UNI) and Brown clusters (BWN) for building the feature space, and we add two new types of features:

1. POS n-grams (POS): the tweet is POS-tagged and the frequency of each POS unigram and bigram is counted.

2. Distant Polarity (DP): two features consisting of the positive and negative probabilities returned by a logistic regression model trained from a distant supervision corpus of 1.6. million tweets labelled with positive and negative emoticons (Go et al., 2009) using unigrams as features. These features are analogous to the soft labels obtained with the model transfer approach in Chapter 3.

The tokenisation process, the POS tags, and the Brown clusters are taken again from the *TweetNLP* project.

We also use the negative sampling method for training skip-gram word-embeddings (W2V) from the target corpus that is implemented in *word2vec*[5]. In this method, a neural network with one hidden layer is trained for predicting the words surrounding a centre word, within a window that is shifted along the target corpus.

The NRC-10 words that occur in the target corpus are labelled according to the corresponding emotions and their feature vectors are used for training a multi-label classifier. We use three multi-label classification techniques:

1. Binary Relevance (BR), in which a separate binary classifier is trained per label.

[5] https://code.google.com/p/word2vec/

2. Classifier Chains (CC) (Read et al., 2011), in which inter-label dependencies are exploited by cascading the predictions for each binary classifier as additional features along a random permutation of labels.

3. Bayesian Classifier Chains (BCC) (Zaragoza, Sucar, Morales, Bielza and Larrañaga, 2011), in which a Bayesian network that represents dependency relations between the labels is learned from the data and used to build a classifier chain based on these dependencies.

The resulting classifiers are used to classify the remaining unlabelled words into emotions.

4.2.2 Evaluation

The proposed approach is evaluated both intrinsically and extrinsically as described in the sub-sections below.

Intrinsic Evaluation

We start with an intrinsic evaluation comparing the micro-averaged and macro-averaged F_1 measures obtained for the ten affective labels. We consider different combinations of features and classifiers. These experiments are carried out using MEKA[6], a toolbox for multi-label classification. In order to obtain association scores for each label we use an L_2-regularised logistic regression from LIBLINEAR, with the regularisation parameter C set to 1.0, as the base learner in the different models.

Multi-label evaluation measures aggregate classification errors of multiples labels. NRC-10 has many words that are not associated with any affective category. Consequently, the majority class for each label is the negative class, which means the corresponding emotion is not present. We use micro and macro F_1 scores to avoid obtaining misleading results from biased models that tend to classify all words to the majority class for all labels, i.e., classify all words as neutral. We describe how to calculate these measures for a particular multi-label classifier as follows.

Let E be the set of all the affective categories ($|E| = 10$), A_i be the words associated with affective category e_i, O_i be the words classified to class e_i, and $B_i = A_i \cap O_i$ be the words that are correctly classified by the model to e_i. As seen in Chapter 2, the precision (P_i), recall (R_i), and F_1 score (F_{1i}) for a single label e_i are calculated as:

[6] http://meka.sourceforge.net/

$$P_i = \frac{B_i}{O_i} \tag{4.8}$$

$$R_i = \frac{B_i}{A_i} \tag{4.9}$$

$$F_{1i} = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}. \tag{4.10}$$

The macro-averaged F_1 is calculated by averaging the F_1 scores of all the affective categories:

$$\text{macroF}_1 = \frac{1}{|E|} \sum_{i=1}^{|E|} F_{1i}. \tag{4.11}$$

This measure treats all the labels as equally important. Hence, it is very sensitive to changes in infrequent labels.

On the other hand, micro-averaged scores take the label distribution into account. Thus, they give more importance to frequent labels (Sintsova and Pu, 2016).

The micro-averaged Precision, Recall, and F_1 score, are calculated according to the following expressions:

$$\text{microP} = \frac{\sum_i |B_i|}{\sum_i |O_i|} \tag{4.12}$$

$$\text{microR} = \frac{\sum_i |B_i|}{\sum_i |A_i|} \tag{4.13}$$

$$\text{microF}_1 = \frac{2 \cdot \text{microP} \cdot \text{microR}}{\text{microP} + \text{microR}}. \tag{4.14}$$

All NRC-10 words that occur at least fifty times in the target corpus are used in our experiments. There were $10,137$ such words (902 are associated with anger, 694 with anticipation, $1,101$ with fear, 579 with joy, 885 with sadness, 432 with surprise, 981 with trust, $2,314$ with negative sentiment, and $1,818$ with positive sentiment).

Before training the word embeddings (W2V) from the target corpus of ten million tweets, we tune the window size and dimensionality of the skip-gram model by conducting a grid-search process in which we train a binary relevance word-level multi-label classifier on the NRC-10 words with 2-fold cross-validation for each parameter configuration. This process is performed over a collection of 1 million tweets independent from the target corpus using the micro averaged F1 measure as performance metric. As shown in the heatmap

in Figure 4.2, the optimum parameters are a window size of 5 and the number of dimensions set to 400. We used this parameter configuration for training the W2V features from the target corpus. From the figure, we can observe that embeddings built using windows smaller than two are not sufficient for capturing emotion-bearing words.

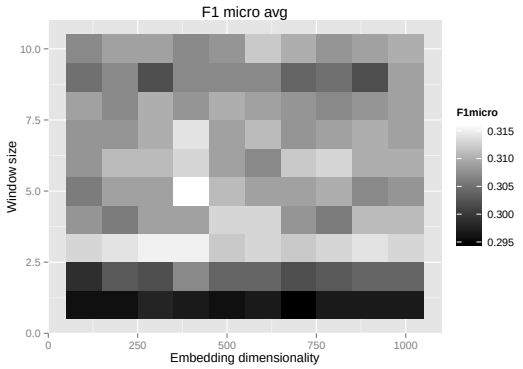

Figure 4.2: Emotion classification results obtained using word embeddings of different dimensionalities, generated from various window sizes. Maximum F1 is achieved for 400 by 5.

The word-level multi-label classification results for the micro-averaged and macro-averaged F_1 measures obtained by training the three multi-label classification schemes BR, CC[7], and BCC using 10-fold cross-validation are shown in Table 4.8. We compare word-level vectors by concatenating different combinations of the features presented in Section 4.2.1: UNI, BWN, POS, DP, and W2V. The unigram feature-space (UNI) is used as the baseline and is compared with the other feature spaces using a corrected resampled paired t-test with an α level of 0.05 (Nadeau and Bengio, 2003).

From the table we can see that distributional features that go beyond word counts, such as BWN, and DP, produce statistically significant improvements over using unigrams alone. On the other hand, W2V alone obtains a better performance than the other features and is only slightly improved when combined with certain features such as DP. This suggests that low-dimensional embeddings trained from unlabelled tweets capture stronger information for emotion classification than word-level features derived by the tweet centroid model. Although these features can produce a competitive representation they do not add much value to W2V. Regarding the multi-label classification techniques, there are no observable benefits of methods that exploit label inter-

[7] Ensembles of classifier chains were also evaluated, with no evidence of improvement over a single chain.

Micro-Averaged F_1			
Classifier	BR	CC	BCC
UNI (Baseline)	0.389 ± 0.03	0.371 ± 0.03	0.378 ± 0.03
UNI-BWN	0.410 ± 0.03 +	0.400 ± 0.03 +	0.407 ± 0.03 +
UNI-BWN-POS	0.411 ± 0.03 +	0.405 ± 0.02 +	0.407 ± 0.03 +
UNI-BWN-POS-DP	0.433 ± 0.03 +	0.427 ± 0.03 +	0.432 ± 0.03 +
UNI-BWN-POS-DP-W2V	0.477 ± 0.03 +	0.474 ± 0.03 +	0.478 ± 0.03 +
W2V	0.473 ± 0.03 +	0.469 ± 0.03 +	0.472 ± 0.03 +
W2V-BWN	0.468 ± 0.03 +	0.469 ± 0.03 +	0.47 ± 0.03 +
W2V-BWN-POS	0.465 ± 0.03 +	0.466 ± 0.03 +	0.466 ± 0.02 +
W2V-BWN-POS-DP	0.474 ± 0.03 +	0.473 ± 0.03 +	0.475 ± 0.03 +
W2V-DP	**0.479** ± 0.03 +	**0.476** ± 0.03 +	**0.479** ± 0.03 +
Macro-Averaged F_1			
Classifier	BR	CC	BCC
UNI (Baseline)	0.272 ± 0.03	0.236 ± 0.02	0.257 ± 0.03
UNI-BWN	0.318 ± 0.02 +	0.302 ± 0.03 +	0.316 ± 0.03 +
UNI-BWN-POS	0.320 ± 0.02 +	0.308 ± 0.02 +	0.319 ± 0.03 +
UNI-BWN-POS-DP	0.344 ± 0.03 +	0.335 ± 0.02 +	0.346 ± 0.02 +
UNI-BWN-POS-DP-W2V	**0.401** ± 0.03 +	0.391 ± 0.03 +	**0.402** ± 0.03 +
W2V	0.392 ± 0.03 +	0.381 ± 0.03 +	0.393 ± 0.03 +
W2V-BWN	0.390 ± 0.02 +	0.388 ± 0.02 +	0.395 ± 0.02 +
W2V-BWN-POS	0.388 ± 0.02 +	0.385 ± 0.02 +	0.392 ± 0.02 +
W2V-BWN-POS-DP	0.398 ± 0.03 +	**0.392** ± 0.03 +	0.401 ± 0.03 +
W2V-DP	0.397 ± 0.03 +	0.391 ± 0.03 +	0.400 ± 0.03 +

Table 4.8: Word-level multi-label classification results. Best results per column for each performance measure are shown in bold. The symbol + corresponds to statistically significant improvements with respect to the baseline.

dependencies, such as CC and BCC, over BR.

The trained multi-label classifiers are used to create Twitter-specific word-emotion associations by classifying the $42,900$ unlabelled words from the corpus into 10-dimensional affect vectors. A word cloud of the expanded lexicon that combines all the features, trained with BCC, is shown in Figure 4.3. The word sizes are proportional to the estimated probabilities associated with the corresponding emotions.

Most of the word-emotion associations shown in the figure are intuitively plausible. However, there also words with prominent associations that are not intuitive, such as the number 17.00 for the emotion surprise. This token receives an association probability of 0.99 for that emotion. We inspected the tweets from the Edinburgh corpus containing this token and we found the following tweet retweeted around 200 times:

- We hope you like our random raffles ? Retweet this msg for an entry into our monday raffle ! Only until 17.00 GMT !

Figure 4.3: A visualisation for the expanded emotion lexicon.

The reason why the token 17.00 exhibits a strong association with surprise is because it co-occurs multiple times with the word raffle, which in turn, is an NRC-10 surprise word. The tweet centroid model used for building this lexicon produces similar vectors for words that are likely to co-occur in a tweet. This type of association is referred to as first-order co-occurrence or syntagmatic association (Schütze and Pedersen, 1993).

This shows that the tweet centroid model can be sensitive to over-represented co-occurrences produced by retweets. In this case the retweets occurred because the author offered a reward for it. A possible approach to tackled this problem is to discard retweets from the corpus, or to discard numbers and other type of words unlikely to bear emotions.

On the other hand, the word 17.00 is not strongly associated with any emotion in the lexicon built using only W2V embeddings. This is because in the skip-gram model words that co-occur in a corpus of tweets do not necessarily produce similar vectors. The skip-gram model is based on second-order co-occurrences or paradigmatic associations (Schütze and Pedersen, 1993), where words need to have similar neighbouring words in order to receive similar vectors.

This example suggests that second-order associations are more robust to over-represented co-occurrences for producing semantic vectors.

Extrinsic Evaluation

We conduct an extrinsic evaluation by studying the usefulness of the expanded lexicons for classifying Twitter messages into emotion categories. We use the Twitter Emotion Corpus (Mohammad, 2012), which has $21,051$ tweets labelled by a single-label multi-class emotional label. The labelling was performed using hashtags. The number of tweets per class is $3,849$ for surprise, $3,830$ for sadness, $8,240$ for joy, 761 for disgust, $2,816$ for fear, and $1,555$ for anger. Using 10-fold cross-validation, we compare a one-vs-all logistic regression that uses attributes calculated from NRC-10 alone (the baseline), with the performance obtained by a classifier trained with attributes derived from NRC-10 and the expanded lexicon.

As previously, the comparisons are carried out using the corrected resampled paired t-test. We calculate ten numerical features from NRC-10 by counting the number of words in a tweet matching each emotion category, and another ten features from the expanded lexicon, calculated as the sum of the corresponding affect probabilities for the matched words, obtained from the multi-label word-level model. Therefore, tweets are represented by ten features in the baseline (NRC-10 alone), and by twenty features for each expanded lexicon (with one lexicon for each multi-label classifier considered above). The kappa statistic and weighted area under the ROC curve (AUC) for all the logistic regression models trained with different expanded lexicons is given in Table 4.9.

Lexicon	Kappa			AUC		
NRC-10 (alone)	0.077			0.633		
NRC-10+Expanded	BR	CC	BCC	BR	CC	BCC
UNI	0.191 +	0.201 +	0.198 +	0.711 +	0.714 +	0.713 +
UNI-BWN	0.174 +	0.178 +	0.176 +	0.708 +	0.712 +	0.711 +
UNI-BWN-POS	0.175 +	0.177 +	0.178 +	0.708 +	0.711 +	0.710 +
UNI-BWN-POS-DP	0.180 +	0.183 +	0.184 +	0.713 +	0.715 +	0.714 +
UNI-BWN-POS-DP-W2V	0.187 +	0.197 +	0.183 +	0.712 +	0.714 +	0.713 +
W2V	**0.223** +	**0.226** +	**0.226** +	0.720 +	**0.723** +	**0.723** +
W2V-BWN	0.199 +	0.201 +	0.197 +	0.713 +	0.715 +	0.715 +
W2V-BWN-POS	0.195 +	0.201 +	0.196 +	0.710 +	0.713 +	0.712 +
W2V-BWN-POS-DP	0.199 +	0.204 +	0.199 +	0.714 +	0.715 +	0.715 +
W2V-DP	**0.223** +	0.223 +	**0.226** +	**0.722** +	**0.723** +	**0.723** +

Table 4.9: Message-level classification results over the Hashtag Emotion Corpus. Best results per column are given in bold.

All the expanded lexicons are statistically significantly better than using NRC-10 alone. Note that all these improvements are substantial in all cases. Similarly to the intrinsic results, we observe that the lexicons created using

W2V alone and W2V-DP are the strongest ones. Another interesting result is that lexicons created with multi-label classifiers that exploit label correlations, such as CC and BCC, are slightly better than the ones created using BR in most cases.

4.3 Discussion

In this chapter, we studied distributional representations for acquiring sentiment knowledge by classifying Twitter words into affective dimensions in a supervised fashion. Our experimental results show the usefulness of the induced words for message-level classification into both polarity and emotion categories. The main advantage of this methodology is that it depends on resources that are relatively cheap to obtain: a seed lexicon, and a collection of unlabelled tweets. The former can be obtained from publicly available resources such as the ones used in this work, and the latter can be freely collected from the Twitter API.

In contrast to earlier work on creating polarity and emotion lexicons for Twitter (Becker et al., 2013; Mohammad et al., 2013; Zhou et al., 2014; Mohammad and Kiritchenko, 2015), which are restricted to tweets annotated with emoticons or emotional hashtags, our methodology can learn affective words from any collection of unannotated tweets. Hence, our approach can be used, without any additional labelling effort, for creating domain-specific emotion lexicons based on unlabelled tweets collected from the target domain, such as politics and sports.

We also observed that low-dimensional word-embeddings are better than distributional word-level features obtained by averaging tweet-level features. This is aligned with recent findings in NLP showing that representations learned from unlabelled data using neural networks outperform representations obtained from hand-crafted features (Baroni, Dinu and Kruszewski, 2014).

The tuning process for the parameters of the W2V embeddings in Figure 4.2 includes the test data for the intrinsic task. This is because all NRC-10 word are used for tuning the embeddings, and they are also used for training and testing the word-level classifiers. However, the embeddings are tuned using vectors calculated from an independent collection of tweets, which are different to the ones used in the intrinsic evaluation. This should mitigate optimistic bias in the intrinsic task. Furthermore, W2V also exhibited strong results in the extrinsic task, which was completely independent of the NRC-10 words. This suggests that the embeddings are unlikely to be overfitted.

Note that the tweet centroid model proposed in this chapter has two novel properties that are lacking from other semantic vector models:

1. It can be used together with any type of message-level feature set.

2. It represents words and tweets in the same vector space.

The first property enables the creation of task-specific word vectors. For example, the DP feature used in this chapter is a sentiment-specific feature obtained from message-level attributes. The second property will be further exploited in Chapter 5, where we will use it for transferring sentiment knowledge between words and tweets.

Chapter 5

Transferring Sentiment Knowledge between Words and Tweets

Thus far we have presented two models for inducing Twitter-specific polarity lexicons and shown that lexicons built with these models can produce useful features for classifying the sentiment of tweets. Nevertheless, sentiment-annotated tweets are required in order to train a message-level polarity classifier that uses these lexicon-based features, and we know from our discussion of the label sparsity problem that obtaining these annotations is time consuming and labour intensive.

Another drawback of these lexicon induction models is that they depend on a seed lexicon of labelled words for training the word-level classifier. These lexicons are not necessarily available for all languages. When no seed lexicon is available, it is desirable to be able to induce a lexicon from sentiment-annotated tweets (e.g., tweets annotated with emoticons), as is done by the PMI-SO method introduced in Chapter 2. In this chapter, we introduce a transfer learning approach for achieving this.

Transfer learning refers to the process of improving the learning of a predictive function for a target domain $\mathcal{D}_{\mathcal{T}}$ using knowledge obtained from a related source domain $\mathcal{D}_{\mathcal{S}}$ (Pan and Yang, 2010). Inspired by this principle, we will use the tweet centroid model (TCM) introduced in Chapter 4 for transferring sentiment knowledge from the word domain $\mathcal{D}_{\mathcal{W}}$ to the message domain $\mathcal{D}_{\mathcal{M}}$ and vice versa.

The tweet centroid model represents tweets and words by feature vectors of the same dimensionality. Tweets can be represented using standard natural language processing (NLP) features such as unigrams and part-of-speech (POS) tags, and words are represented by the centroids of the tweet vectors in which they occur.

Classifiers trained from the word or message domain can be deployed on

data from the other domain because both tweets and words can be labelled according to the same sentiment categories, e.g, positive and negative ($\mathcal{Y}_\mathcal{W} = \mathcal{Y}_\mathcal{M}$). Therefore, a word-level classifier trained from a polarity lexicon and a corpus of unlabelled tweets can be used for classifying the sentiment of tweets. Likewise, we can train a message-level classifier from a corpus of sentiment-annotated tweets and use it for classifying words into sentiment classes. This idea is illustrated in Figure 5.1.

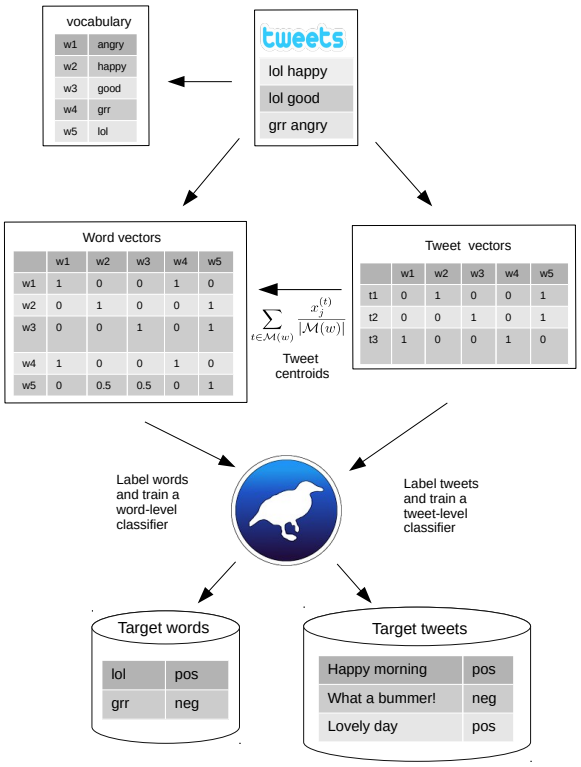

Figure 5.1: Transfer Learning with tweet centroids. The bird represents the Weka machine learning software.

This transfer learning approach is useful in scenarios where either message-level polarity classification or polarity lexicon induction needs to be performed but it is easier to obtain annotated data from the other domain. We evaluate our approach in two transfer learning problems: 1) training a tweet-level polarity classifier from a polarity lexicon, and 2) inducing a polarity lexicon from a collection of polarity-annotated tweets. Our results show that the proposed approach can successfully classify words and tweets after transfer.

The transferability of sentiment knowledge between words and tweets is based on the hypothesis that there is a sentiment interdependence relation between them. This relation, which was first observed in (Sindhwani and

Melville, 2008) in the case of larger text documents, is defined by the following two statements:

1. The polarity of a tweet is determined by the polarity of the words it contains.

2. The polarity of a word is determined by the polarity of the tweets in which it occurs.

The remainder of this chapter is organised as follows. The proposed transfer learning approach is formalised in Section 5.1. In Section 5.2, we present the experiments we conducted to evaluate the proposed approach and discuss results. The main findings are discussed in Section 5.3.

5.1 Tweet-Centroids for Transfer learning

In this section we revisit the tasks of message-level polarity classification and polarity lexicon induction. Afterwards, we show how the tweet centroid model can be used for sentiment transfer learning between words and messages.

Following the notation proposed in (Pan and Yang, 2010), a domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$ and each x_i is a numeric feature. Given a particular domain $\mathcal{D}$, a task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a predictive function f that can be learned from training data consisting of pairs $\{x, y\}$ where $x \in X$ and $y \in \mathcal{Y}$. The function f can be used for predicting the corresponding label $f(x)$ of a new instance x.

In the Twitter sentiment analysis context, a tweet or message m is formed by a sequence of words. We assume a tweet is represented by a k-dimensional vector $\overrightarrow{x}$ residing in a feature space $\mathcal{X}_{\mathcal{M}}$ that belongs to the message domain $\mathcal{D}_{\mathcal{M}}$. Different NLP features can be used to form $\mathcal{X}_{\mathcal{M}}$. In this chapter, we again consider three types of features that have proven to be useful for sentiment analysis of tweets (Kiritchenko et al., 2014): 1) Word unigrams (UNI), 2) Brown clusters (BWN), and 3) part-of-speech tags (POS).

The message-level sentiment label space $\mathcal{Y}_{\mathcal{M}}$ corresponds to the different sentiment categories that can be expressed in a tweet, e.g., positive, negative, and neutral. For simplicity, we will only consider the two classes positive and negative in this chapter. Because sentiment is a subjective judgment, the ground-truth sentiment category of a tweet must be determined by a human evaluator.

Given a corpus of sentiment-annotated tweets $\mathcal{C}_L$, a message-level polarity classifier f_M can be trained using standard supervised learning methods and then be used for the message-level polarity classification task. Annotated corpora are commonly not available for creating domain-specific sentiment classifiers due to the high costs involved in the annotation process. On the other hand, a large corpus of unlabelled public tweets $\mathcal{C}_U$ can be freely obtained from the Twitter API. Tweets restricted to a specific language, geographical region, or set of key words can also been collected for creating domain-specific collections.

Words can be annotated according to the same sentiment categories as messages ($\mathcal{Y}_\mathcal{W} = \mathcal{Y}_\mathcal{M}$) to indicate their prior sentiment. Examples of positive words are *happy* and *great*, and examples of negative ones are *sad* and *miserable*. Again, the ground-truth sentiment of a word is a subjective judgment determined by a human. We refer to a list of words annotated by sentiment as a polarity lexicon $\mathcal{L}$.

The tweet centroid model proposed in Chapter 4 represents words as the centroids of the k-dimensional tweet vectors pertaining to those tweets in a corpus of unlabelled tweets $\mathcal{C}_U$ that contain the words. Thus, the tweet centroid model can be used to form a word domain $\mathcal{D}_\mathcal{W}$ with the same feature space as the one used for representing the messages ($\overrightarrow{w} \in \mathcal{X}$) in the message domain $\mathcal{D}_\mathcal{M}$.

Taking the words from the vocabulary that match a given polarity lexicon ($\mathcal{V} \cap \mathcal{L}$), a word-level polarity classifier f_W can be trained and used for classifying the remaining unlabelled words, thus solving the polarity lexicon induction task as done in Chapter 4.

Transfer learning requires the source and the target tasks to be related to each other. We hypothesise that there is a strong relationship between message-level polarity classification and polarity lexicon induction because the sentiment of a tweet is associated with the sentiment of the words it contains and the sentiment of a word is associated with the sentiment of the tweets that use it.

Assuming that this hypothesis is true, we can apply the tweet centroid model for addressing message-level polarity classification and polarity lexicon induction by taking labels from the respective other domain. Considering that both tweets and words reside in the same feature space, and given a collection of unlabelled tweets $\mathcal{C}_U$, we can classify the sentiment of messages using a word-level classifier f_W trained with tweet centroids labelled by a polarity lexicon

$\mathcal{L}$.

It is important to note that the number of labelled words for training f_W is limited to the number of words from $\mathcal{L}$ occurring in $\mathcal{C}_U$. As was shown in Chapter 2, most of existing hand-annotated polarity lexicons contain less than $10,000$ words. This means that our method is not capable of exploiting large collections of unlabelled tweets for producing training datasets larger than the size of $\mathcal{L}$. We propose a modification of our method for increasing the number labelled instances it produces. The modification is based on partitioning the word-tweet sets. The word-tweet set $\mathcal{M}(w)$ for each word from the lexicon ($w \in \mathcal{L}$) is partitioned into smaller disjoint subsets $\mathcal{M}(w)_1, \ldots \mathcal{M}(w)_z$ of a fixed size determined by a parameter p where $z = |M(w)|/p$. We calculate one tweet centroid vector $\overrightarrow{w}$ for each partition labelled according to $\mathcal{L}$. As is shown in Section 5.2.2, this modification leads to substantial improvements when transferring sentiment knowledge from words to tweets.

The reverse transfer of sentiment knowledge is also possible. Given a message-level polarity classifier f_M trained on a corpus of tweets $\mathcal{C}_L$ annotated by sentiment, a polarity lexicon can be induced by applying f_M to the words in $\mathcal{C}_L$, simply by representing these words by the centroids of the tweets in C_L that contain them. Alternatively, considering that sentiment-annotated corpora are usually small and word-level distributional representations such as these centroids capture richer semantic information when calculated from large document corpora, it is also possible to perform the induction by applying f_M to word vectors (i.e., tweet centroids) calculated from a larger corpus of unlabelled tweets $\mathcal{C}_U$.

Our transfer learning approach is novel in the sense that both the source domain and target domain are represented with the same feature space ($\mathcal{X}_\mathcal{M} = \mathcal{X}_\mathcal{W}$). In most previous transfer learning models for text classification the features spaces of the two domains are different (Pan and Yang, 2010).

It is important to clarify that the message domain $\mathcal{D}_\mathcal{M}$ and the word domain $\mathcal{D}_\mathcal{W}$ do not have the same probability distribution and, hence, our model performs transfer learning according to the definition from Pan and Yang (2010). The probability distribution of the tweet domain, $P(X_m)$, is formed by sparse features such as unigrams and Brown clusters, whereas the distribution of the word domain, $P(X_w)$, is formed by averaging vectors from the tweet domain, which yields dense vectors with lower variance. Moreover, the conditional distributions of the two sentiment classification tasks are not the same either. $P(Y_w|X_w)$ encodes the relation between the prior polarity of a word and its

distributional representation, whereas $P(Y_m|X_m)$ represents the relation between the polarity of a message and its sparse feature vector. Hence, normally, $P(Y_w|X_w) \neq P(Y_m|X_m)$[1].

The two domains are clearly different. However, assuming that the sentiment interdependence relation is true, we expect the two domains to be sufficiently associated with each other to allow the transferability of sentiment knowledge between them.

5.2 Experiments

In this section, we conduct an experimental evaluation of the proposed approach. The evaluation is divided into three parts. First, we empirically study the interdependence relation between tweets and words. Second, we evaluate how to transfer sentiment labels from words to tweets. Finally, we evaluate how to induce a polarity lexicon from tweets annotated by sentiment.

5.2.1 The word-tweet sentiment interdependence relation

We start by studying the sentiment interdependence relation between documents and words in Twitter: the sentiment of documents determines the sentiment of the words they contain, while the sentiment of words determines the sentiment of the tweets that contain them.

We describe positive and negative tweets based on the polarity of their words, and likewise, describe positive and negative words from a given polarity lexicon according to the polarity of the tweets in which they occur. We expect to observe clear differences between elements of different polarities based on these descriptions. The annotated data we use for this is taken from the *SemEval* corpus of sentiment annotated tweets and the AFINN lexicon (Årup Nielsen, 2011) of positive and negative words. Both datasets were already used in previous chapters.

The *SemEval* (Nakov et al., 2013) corpus is formed by 5232 positive tweets and 2067 negative tweets annotated by human evaluators using the crowdsourcing platform Amazon Mechanical Turk[2]. Each tweet is annotated by five Mechanical Turk workers and the final label is determined based on the majority of the labels.

[1] If we consider the partitioned version of the model, the smaller the value of the partition size p, the more similar the conditional distributions of the two domains. Indeed, if p is set to one, both distributions are the same.

[2] http://www.mturk.com

The AFINN lexicon is formed by 1176 positive words and 2204 negative words, annotated by Finn Årup Nielsen, and includes informal words commonly found in Twitter such as slang, obscene words, acronyms and Web jargon. AFINN does not include any emoticons.

We describe each tweet from *SemEval* by a message-level polarity variable calculated as the difference between the number of positive and negative words from the AFINN lexicon found in the message. This variable is normalised by the total number of words in the tweet. The tweets that do not have words from the lexicon are discarded, resulting in 1638 negative and 4193 positive tweets. The median of this variable for negative and positive tweets is -0.04 and 0.05 respectively. The polarity of positive and negative categories is also compared using a Wilcoxon rank sum test obtaining a p-value less than $2.2e^{-16}$. This shows that there is statistical evidence that negative tweets are more likely to be formed by negative words than positive ones, and likewise positive tweets are more likely to contain positive words than negative ones. These results support the first part of the proposed tweet-word sentiment interdependence relation: the sentiment of a tweet is determined by the polarity of its words.

We also describe each word from the AFINN lexicon by a word-level polarity variable calculated as the difference between the number of positive and negative tweets that contain it. This variable is normalised by the total number of tweets in which the word is used. To reduce the noise induced by infrequent words, we discard words occurring in fewer than three tweets, resulting in 259 positive and 250 negative words. The median of the word-level polarity for positive and negative classes is 0.76 and -0.33 respectively. We compare this variable for both sentiment classes using a Wilcoxon rank sum test and the resulting p-value is again less than $2.2e^{-16}$. Hence there is also statistical evidence that positive and negative words occur more frequently in tweets with the same polarity than in tweets with the opposite one. These results support the second part of the tweet-word sentiment interdependence relation: the sentiment of a word is determined by the sentiment of the tweets in which it occurs.

The distribution of the message-level and word-level polarity variables for each corresponding sentiment category is shown in the violin plots in Figure 5.2.

From the plot we can observe that the interquartile range of the tweet-level polarity lies below zero for the negative class and above zero for the posi-

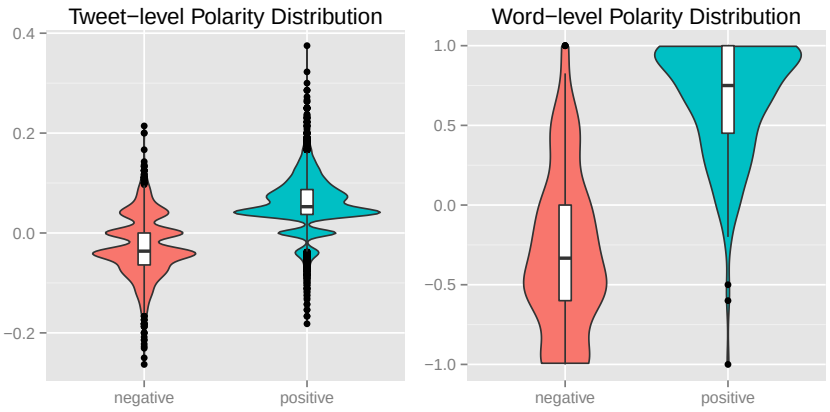

Figure 5.2: Violin plots of the polarity of tweets and words.

tive one, suggesting that tweets of different sentiment classes have different distributions when considering the sentiment of their words. Regarding the words we can again observe that the interquartile ranges lie below and above zero for negative and positive words respectively. Note that the gap between the positive and negative interquartile range is larger than the corresponding gap in the case of tweets. We believe that this is because there is more complete information available for describing words according to the polarity of the tweets in which they occur than for describing tweets according to the polarity of their words. In one case, the sentiment labels of the tweets in which opinion words occur are fully given by the sentiment-annotated corpus. In the other case, we only have the polarity of the words from a tweet that match the lexicon but do not have sentiment information for the other words in the tweet.

5.2.2 From opinion words to sentiment tweets

In this subsection, we evaluate the transfer of sentiment labels from words to tweets for solving the message-level polarity classification task. We train a word-level classifier f_W on tweet centroids calculated from a collection of unlabelled tweets $\mathcal{C}_U$, where these centroids are labelled according to a polarity lexicon $\mathcal{L}$. We also study the effect of partitioning the word-tweet sets to increase the number of training instances obtained with our tweet centroid method.

The collection of unlabelled tweets is taken again from the Edinburgh corpus and we use AFINN as the polarity lexicon for the centroid labels.

The features used for representing the tweets and the words from $\mathcal{C}_U$ are:

unigrams, POS tags, and Brown clusters. The tweets are lowercased, and user mentions and URLs are replaced by special tokens in the same way as in previous chapters.

We only consider word vectors of words that are included in the lexicon, and we also discard words occurring in fewer than ten tweets to avoid learning spurious relationships from infrequent words. The classifier is trained using an L_2-regularised logistic regression taken from LIBLINEAR, with the regularisation parameter C set to 1.0.

We compare our model with classifiers trained using two distant supervision methods for obtaining training instances from unlabelled corpora: the emoticon-annotation approach (EAA) and the lexicon-annotation approach (LAA).

In EAA we use the following positive and negative emoticons for labelling tweets from the source collection: ":)", ":D", "=D", "=)", ":]", "=]", ":-)", ":-D", ":-]", ";)", ";D", ";]", ";-)", ";-D", and ";-]" for positive tweets and ":(", "=(", ";(", ":[", "=[", ":-(", ":-[", ":'(", ":'[", and "D:" for negative tweets. Tweets without emoticons and tweets containing both positive and negative emoticons are discarded. The emoticons are removed from the content after labelling.

In LAA the tweets from $\mathcal{C}_U$ are labelled using the AFINN lexicon. The tweets with at least one positive word and no negative word are labelled positive, and analogously, tweets with at least one negative word and no positive word are labelled negative.

The classification of tweets into sentiment categories using a model trained from word vectors represented by the tweet centroid model is a form of distant supervision because we are relying on a heuristic function for automatically obtaining training data for our message-level sentiment classification task. The goal of comparing TCM against EAA and LAA is to determine which distant supervision model generates better training data for polarity classification from a source corpus of unlabelled tweets.

It is important to recall that the training examples produced with the three methods reside in the same feature space. We study different configurations of TCM. The first configuration is the original version of TCM, in which we obtain one instance per word. The other configurations correspond to partitioned versions of TCM, in which the tweet-word sets of each word from the lexicon are randomly partitioned into disjoints subsets of size p. The centroids are calculated from the partitions, and hence, multiple training instances are produced for words occurring in more than p tweets. The partitioning is implemented by enumerating the tweets in each word-tweet set and creating

consecutive sublists of size p. The last partition of the set will be smaller than p if there is a remainder when dividing the size of the set by the value of p.

The evaluation of the classifiers is carried out on the same three manually annotated collections of tweets used in Chapters 3 and 4: *SemEval*, *6HumanCoded*, and *Sanders*. The target tweets are represented by the same features as the tweets used for building the word vectors.

As was described in the previous subsection, the *SemEval* corpus is formed by 5232 positive and 2067 negative hand-annotated tweets. The *6HumanCoded* dataset is a collection of tweets scored according to positive and negative numeric scores by at least three human evaluators. The ratings are averaged and we use the difference of these scores to create polarity classes and discard messages where this difference is zero. The resulting dataset has 1340 positive and 949 negative tweets. The *Sanders* dataset consists of 570 positive and 654 negative tweets evaluated by a single human annotator.

We study the average performance obtained by classifiers trained on labelled instances generated by different configurations of TCM, EAA, and LAA, using ten independent subsamples of 2 million tweets from the Edinburgh corpus as the source data. The average number of positive and negative instances obtained by each model from the ten subsamples is shown in Table 5.1.

We can see from the table that LAA produces the largest training dataset and that the original version of TCM produces the smallest one. Regarding the partitioned version of TCM, we observe that the lower the value of p, the larger the number of instances produced.

From the ten training sets, we compare the average area under the ROC curve (AUC) obtained on the three target collections of tweets for TCM and the two baselines EAA and LAA using a paired Wilcoxon signed-rank test with the significance value set to 0.05. AUC is a useful metric for comparing the performance of classifiers because it is independent of any specific value for the decision threshold. The comparisons are done for the three target collections of tweets and the results are given in Table 5.2. The statistical significance tests of each configuration of TCM with respect to EAA and LAA are indicated by a sequence of two symbols. Improvements are denoted by a plus (+), degradations by a minus (-), and cases where no statistically significant difference is observed by an equals (=). The baselines are also compared against each other.

Regarding the baselines, we observe that LAA is better than EAA in *6HumanCoded* and *SemEval* but worse in *Sanders*. The original version of TCM is

	Avg. Positive	(%)	Avg. Negative	(%)	Avg. Total	(%)
EAA	130,641	(6.5%)	21,537	(1.1%)	152,179	(7.6%)
LAA	681,531	(34.1%)	294,177	(14.7%)	975,708	(48.8%)
TCM	1537	(0.05%)	951	(0.08%)	2488	(0.12%)
TCM (p=5)	276,696	(13.8%)	149,989	(7.5%)	426,684	(21.3%)
TCM (p=10)	138,596	(6.9%)	75,390	(3.8%)	213,986	(10.7%)
TCM (p=20)	69,518	(3.5%)	38,044	(1.9%)	107,563	(5.4%)
TCM (p=50)	32,231	(1.6%)	17,950	(0.9%)	50,181	(2.5%)
TCM (p=100)	14,338	(0.7%)	8357	(0.4%)	22,695	(1.1%)

Table 5.1: Average number of positive and negative instances generated by different models from 10 collections of 2 million tweets.

	6HumanCoded		Sanders		SemEval	
EAA	0.805 ± 0.005	= -	0.800 ± 0.017	= +	0.802 ± 0.006	= -
LAA	0.809 ± 0.001	+ =	0.778 ± 0.002	- =	0.814 ± 0.000	+ =
TCM	0.776 ± 0.004	- -	0.682 ± 0.024	- -	0.779 ± 0.008	- -
TCM (p=5)	0.834 ± 0.002	+ +	0.807 ± 0.008	= +	0.833 ± 0.002	+ +
TCM (p=10)	0.845 ± 0.003	+ +	**0.817** ± 0.006	+ +	0.841 ± 0.002	+ +
TCM (p=20)	**0.850** ± 0.003	+ +	0.815 ± 0.011	+ +	**0.844** ± 0.003	+ +
TCM (p=50)	0.844 ± 0.004	+ +	0.785 ± 0.010	- +	0.836 ± 0.004	+ +
TCM (p=100)	0.829 ± 0.003	+ +	0.752 ± 0.019	- -	0.821 ± 0.004	+ +

Table 5.2: Message-level Polarity Classification AUC values. Best results per column are given in bold.

statistically significantly worse than the two baselines. We believe that this is because non-partitioned TCM generates too few training instances (Table 5.1). In contrast, the partitioned TCM achieves statistically significantly improvements over the two baselines in the three datasets when p equals 10 and 20. We also observe a degradation in performance when the value of p is decreased further (p=5). This suggests a trade-off in the value of p. If p is too large, TCM will generate too few training instances, and conversely, if p is too small, the instances will be calculated by averaging very few tweets, and the resulting distributional word vectors will lack contextual information.

Regarding the performance on the different datasets, we observe a lower performance for *Sanders* in comparison to the other two datasets. Considering that this is the only dataset in which labels are not obtained by averaging multiple human evaluations, we believe that this dataset contains noisier sentiment labels because it reflects the subjective judgement of a single evaluator.

The results obtained in this subsection indicate that opinion words can be successfully transferred to the message level using tweet centroids when the centroids are obtained from partitioned data. Additionally, we conclude that the partitioned tweet centroid method is capable of extracting better informa-

tion from unlabelled tweets than EAA and LAA.

5.2.3 From tweets to opinion words

The research question evaluated in this subsection is whether it is possible to transfer the sentiment knowledge obtained from a sentiment-annotated corpus of tweets for inducing a polarity lexicon. To address this question, we train a message-level classifier f_W from a corpus of sentiment annotated tweets $\mathcal{C}_L$ and deploy it on words found in a corpus of unlabelled tweets, where the words are represented by tweet centroids. We need to have a single instance per word, so we do not partition the word-tweet sets here.

Instead of calculating the target tweet centroids from $\mathcal{C}_L$, we calculate them from a larger corpus of unlabelled tweets $\mathcal{C}_U$ that corresponds to one of the collections of 2 million tweets used in the previous subsection. This is done because of the following reasons:

1. There is empirical evidence that distributional semantic models of words tend to generalise better when calculated from large corpora (Mikolov et al., 2013).

2. By classifying the words from a larger corpus of unlabelled tweets we can induce the polarity of words that do not necessarily occur in the annotated corpus.

As training data for the tweet-level classifier, we use the three annotated collection of tweets that were previously used as testing data for training three message-level classifiers: *Sanders*, *6HumanCoded*, and *SemEval*. We build the feature space with the same features used before: unigrams, POS tags, and Brown clusters. We also use an L_2-regularised logistic regression model with the same parameters for learning the classifier. We only consider labelled words from the AFINN lexicon for evaluation purposes.

We compare the word-level AUC of a message-level classifier deployed on words represented by TCM with the AUC obtained by PMI semantic orientation (PMI-SO) (Turney, 2002), a popular method for inducing polarity lexicons from a corpus of polarity annotated tweets $\mathcal{C}_L$. As was discussed in Chapter 2, PMI-SO corresponds to the difference between the PMI of a word with positive tweets and the PMI of the same word with negative tweets.

The words classified by TCM and PMI-SO are not necessarily the same. TCM classifies words from a larger corpus of unlabelled tweets $\mathcal{C}_U$ rather than clas-

sifying the words from $\mathcal{C}_L$. Therefore, the words induced by TCM are independent of the words in $\mathcal{C}_L$. On the other hand, PMI-SO classifies the words in the labelled corpus $\mathcal{C}_L$. In order to produce a fair comparison between TCM and PMI-SO, we compare the classification performance obtained for the words from AFINN that are classified by both methods. The number of positive and negative words classified by PMI-SO for each source corpus, the number of words classified by TCM for $\mathcal{C}_U$, and the number of words in the intersection, are all shown in Table 5.3.

Set of Words	Pos	Neg	Total
PMI-SO (SemEval)	522	617	1139
PMI-SO (Sanders)	196	231	427
PMI-SO (6HumanCoded)	333	352	685
TCM	961	1554	2515
PMI-SO (SemEval) ∩ TCM	517	602	1119
PMI-SO (Sanders) ∩ TCM	194	227	421
PMI-SO (6HumanCoded) ∩ TCM	332	349	681

Table 5.3: Number of positive and negative words from AFINN.

The AUC values for the intersection of words classified by both PMI-SO and TCM are displayed in Table 5.4. From the table we can observe that TCM outperforms PMI-SO for inducing polarity lexicons when trained on any of the three collections of sentiment annotated tweets. This is a noteworthy result, considering that PMI-SO is a widely-used approach for lexicon induction. We can also observe that classifiers trained from *6HumanCoded* and *SemEval* achieve satisfactory results on the AFINN words. We observe substantially lower performance for the classifier trained from *Sanders*.

AUC		
Source Dataset	PMI-SO	TCM
Sanders	0.757	**0.864**
6HumanCoded	0.861	**0.930**
SemEval	0.858	**0.916**

Table 5.4: Word-level polarity classification results for the AFINN lexicon. Best results per row are given in bold.

These results suggest that the performance of the tweet centroid model for transferring sentiment knowledge from tweets to words can vary substantially depending on the quality of the corpus of sentiment-annotated tweets. We observe that corpora in which the labels are obtained by averaging the judg-

ments of multiple annotators such as *6HumanCoded* and *SemEval* are preferable to corpora annotated by one single individual such as *Sanders*. The size of the corpus could also be a relevant factor, considering that *Sanders* is the smallest collection. It is worth mentioning that when an appropiate source corpus is used, the word-level performance obtained after transfer (Table 5.4) can be even better than for the reverse transfer learning task (Table 5.2).

The probabilistic output of the logistic regression model applied to tweet centroids can be used to explore the sentiment intensities or semantic orientations of Twitter words. We calculate the log odds ratio of the positive and negative probabilities returned by the logistic regression model ($\log_2(\frac{P(pos)}{P(neg)})$) for all the words found in the corpus of unlabelled tweets (here we also include words that are not part of AFINN). In this way, we obtain a sentiment score for each word in which the polarity and the intensity of a word are determined by the sign and the absolute value of the score, respectively.

In Figure 5.3, we use word clouds to visualise the sentiment intensities of positive and negative words classified with the message-level classifier trained from the SemEval dataset.

Figure 5.3: Word clouds of positive and negative words obtained from a message-level classifier.

The left-side word cloud corresponds to positive words in which the log odds are greater than zero ($\log_2(\frac{P(pos)}{P(neg)}) > 0$) and the size of each word is proportional to its log odds score. Analogously, in the right-side word cloud, we show negative words in wich the score is less than zero and the size of the words is proportional to the score multiplied by -1. We observe from the figure that the word-level sentiment intensities transferred from message-level sentiment knowledge are plausible.

5.3 Discussion

In this chapter, we have presented a transfer learning model for transferring sentiment knowledge between words and tweets. This was achieved by representing both tweets and words with the same features and deploying classifiers trained from one domain on data from the other one[3]. A noteworthy aspect of this approach is its simplicity; yet, despite its simplicity, it yields promising classification performance.

We studied the word-tweet sentiment interdependence relation on which the proposed tweet centroid model is based, showing that the sentiment of tweets is strongly related to the sentiment of their words and that the sentiment of a word is strongly related to the sentiment of the tweets in which it occurs.

We observed that the partitioned version of the tweet centroid model allows for accurate classification of the sentiment of tweets using a word-level classifier trained from a corpus of unlabelled tweets and a polarity lexicon of words. The partitioned tweet centroid model (with an appropiate partition size) outperformed the classification performance of the popular emoticon-based method for data labelling and also produced better results than a classifier trained from tweets labelled based on the polarity of their words (LAA). The partitioned tweet centroid model is a lexicon-based distant supervision model that can be used for training message-level classifiers when no tweets annotated by sentiment are available. It can also be used for domains in which emoticons are not frequently used. Considering that opinion lexicons are usually easier to obtain than corpora of sentiment-annotated tweets, the tweet centroid model can significantly reduce cost when solving the message-level polarity classification problem.

Our results also show the feasibility of the reverse transfer process, where a polarity lexicon is induced by applying a message-level polarity classifier. We found that TCM produces more accurate lexicons than the well-known PMI-SO measure. The quality of the induced lexicon depends on the quality and size of the sentiment-annotated Twitter data. An important aspect of TCM for lexicon induction is that the word centroids can be calculated from any collection of unlabelled tweets. Hence, the method can be used for creating domain-specific opinion lexicons by collecting tweets associated with the target domain.

[3]The source code of the model is available for download at http://www.cs.waikato.ac.nz/ml/sa/ds.html#ptcm.

Chapter 6

Lexicon-based Distant Supervision: Annotate-Sample-Average

Distant supervision models such as the emoticon-annotation approach are popular solutions for training message-level polarity classifiers in the absence of sentiment-annotated tweets. A lexicon-based distant supervision model is a particular type of distant supervision approach that exploits prior lexical knowledge in the form of opinion lexicons.

Polarity lexicons are normally formed by thousands of frequently used words, so there is a high probability that a tweet contains at least one word from the lexicon. This means means that lexicon-based distant supervision models can potentially exploit more unlabelled data than the well known emoticon-annotation approach because the latter is based on a small number of positive and negative emoticons.

The partitioned tweet centroid model proposed in Chapter 5 was shown to be an effective model of this type. In this chapter we propose another lexicon-based distant supervision method called *Annotate-Sample-Average* (ASA). ASA takes a collection of unlabelled tweets and a polarity lexicon composed of positive and negative words and creates synthetic labelled instances for Twitter polarity classification. Each labelled training instance is created by sampling with replacement a number of tweets containing at least one word from the lexicon with the desired polarity, and averaging the feature vectors of the sampled tweets. This allows the usage of any kind of features for representing the tweets, e.g., unigrams and part-of-speech tags (POS) tags.

The intuition behind ASA is that a tweet containing a word with a certain known positive or negative polarity has a certain likelihood of expressing the same polarity in the whole message. Of course, the opposite polarity may also be expressed due to the presence of negation, sarcasm, or other opinion words with the opposite polarity. We propose a hypothesis, which we refer to as the

"lexical polarity hypothesis", stating that the first scenario is more likely than the second one. Based on that, when sampling and averaging multiple tweets exhibiting at least one word with the desired positive or negative polarity, we increase the confidence of obtaining a vector located in the region of the desired polarity.

Most sentiment analysis datasets are imbalanced in favor of positive examples (Li et al., 2011). This is presumably because users are more likely to report positive than negative opinions (Guerra et al., 2014). The shortcoming of training sentiment classifiers from imbalanced datasets is that many classification algorithms tend to predict test samples as the majority class (Japkowicz and Stephen, 2002) when trained from this type of data. A popular way to address this problem is to rebalance the data by under-sampling the majority class or by over-sampling the minority class. A noteworthy property of ASA is that it incorporates a rebalancing mechanism in which balanced training data can be generated.

We compare classifiers trained with ASA against the same distant supervision baselines used in Chapter 5: the emoticon-annotation approach and a simple lexicon-based annotation approach that annotates tweets according to the polarity of their words. The experimental results show that ASA, with appropriate choice of the number of tweets averaged for each generated instance, outperforms the other methods in all cases and obtains similar results to the partitioned tweet centroid model from Chapter 5.

This chapter is organised as follows. In Section 6.1, we describe the proposed ASA method. We discuss the differences between ASA and the tweet centroid model in Section 6.2. The lexical polarity hypothesis is empirically studied in Section 6.3. The evaluation of the method is presented in Section 6.4. The main results of this chapter are discussed in Section 6.5.

6.1 The Annotate-Sample-Average Algorithm

In this section, we describe the Annotate-Sample-Average (ASA) algorithm for generating training data for Twitter polarity classification. The method receives two data inputs: 1) a source corpus, and 2) an opinion lexicon.

The source corpus is a collection of unlabelled tweets $\mathcal{C}$ on which the generated instances are based. The corpus can be built using the public Twitter API[1], which allows the retrieval of public tweets. The tweets must be written

[1] https://dev.twitter.com/overview/api

in the same language as the opinion lexicon, and the type of tweets included in the collection should depend on the type of sentiment classifier intended to be built. For instance, in order to build a domain-specific sentiment classifier (e.g., for a political election), the collection should be restricted to tweets associated with the target domain. This can be done using the Twitter API by specifying key words, users, or geographical areas. In the following, we focus on domain-independent polarity classification. Thus, we consider a general purpose collection of English tweets.

The opinion lexicon $\mathcal{L}$ is a list of words labelled by sentiment. In this chapter, we consider positive and negative words. The positive and negative subsets of the lexicon are denoted by symbols $\mathcal{L}_+$ and $\mathcal{L}_-$ respectively. Several existing opinion lexicons can be used here. As was described in Chapter 2, there are basically two families of lexicons that can be considered:

1. Manually annotated lexicons, in which the sentiment of the words is annotated according to human judgements. Crowdsourcing tools such as Amazon Mechanical Turk can be used to support the annotation (Mohammad and Turney, 2013).

2. Automatically-annotated lexicons that are created by automatically expanding a small set of opinion words using relations provided by semantic networks, e.g., synonyms, and antonyms (Kim and Hovy, 2004), or using statistical associations calculated from document corpora, e.g., point-wise mutual information (Turney, 2002).

We observed in Chapter 2 that manually-annotated lexicons tend to be smaller than the automatically made ones. Conversely, automatically-annotated lexicons are likely to be noisy and may include several neutral words that are not very useful for polarity classification. In this chapter we use AFINN, the manually-annotated lexicon that was also used in Chapter 5.

The other parameters of ASA are: a, which determines the number of tweets to be averaged for each generated instance, p, which corresponds to the number of positive instances to be generated, n, corresponding to the number of negative instances, and m, which is a flag specifying how to handle tweets that contain both positive and negative words.

The tweets from $\mathcal{C}$ are preprocessed in the same way as in previous chapters, i.e, tweets are lowercased, and user mentions and URLs are replaced by special tokens.

The first step of the algorithm is the *annotation* phase, in which the tweets from $\mathcal{C}$ are annotated according to the prior sentiment knowledge provided by the lexicon. Every time a positive word from $\mathcal{L}_+$ is found in a message, the whole tweet is added to a set called *posT*; analogously, if a negative word is found in $\mathcal{L}_-$, the tweet is added to a set called *negT*. Tweets with both positive and negative words will be discarded if the flag m is set, and will be simultaneously added to both *posT* and *negT* otherwise.

The tweets contained in *posT* and *negT* are candidates for building synthetic labelled instances for training a tweet classifier. The assumption here is that tweets in each set, positive and negative, are more probable to express the corresponding polarity than the opposite polarity. This can be explained by the short length of tweets. As tweets are short straight-to-the-point messages, the presence of a polarity word has a strong correlation with the overall polarity expressed in the message. For example, the tweet: *"Hey guess what? I think you're awesome"* contains the word *awesome* and is clearly expressing a positive sentiment. Obviously, there are also tweets with opinion words than can express the opposite polarity, e.g., *"Not happy where I'm at in life"*. This can occur due to several factors such as the presence of other words with the opposite polarity, negations, or sarcasm. However, we hypothesise that the first situation is more likely than the second one. We refer to this hypothesis as the "lexical polarity hypothesis" and we study it empirically in Section 6.3.

We represent all the candidate tweets by vectors of features. We consider the same features that we used in Chapter 5: 1) Word unigrams (UNI), 2) Brown clusters (BWN), and 3) Part-of-speech tags (POS).

The second step of ASA is the *sampling* step. ASA randomly samples with replacement a tweets from either *posT* or *negT*. Next, in the *averaging* step, the feature vectors of the sampled tweets are averaged and labelled according to the polarity of the set from which they were sampled. The rationale behind this step is that, assuming that the "lexical polarity hypothesis" holds, averaging multiple tweets sampled from the same set increases the confidence of generating training instances for the tweet classifier that are located in the region of the desired polarity.

We define the random variable D as the event of sampling a tweet with the desired positive or negative polarity from either *posT* or *negT*. We assume that D is distributed with a Bernoulli distribution specified by a parameter p_d[2]. We define another random variable M as the event that the majority of the a ran-

[2]This parameter is greater than 0.5 if the lexical polarity hypothesis holds.

Algorithm 6.1.1: ASA algorithm

```
1  Algorithm ASA(C, L, a, p,n,m)
2      foreach tweet ∈ C do
3          if m and (hasWord(tweet,L+) and hasWord(tweet,L−)) then
4              continue
5          if hasWord(tweet,L+) then
6              tweetVec ← extractFeatures(tweet)
7              posT.put(tweetVec)
8          if hasWord(tweet,L−) then
9              tweetVec ← extractFeatures(tweet)
10             posN.put(tweetVec)
11     end
12     i ← 0
13     while i ≤ p do
14         pInst ← sampleAndAverage(posT,a)
15         pInst.label ← pos
16         O.put(pInst)
17         i ← i + 1
18     end
19     i ← 0
20     while i ≤ n do
21         nInst ← sampleAndAverage(negT,a)
22         nInst.label ← neg
23         O.put(nInst)
24         i ← i + 1
25     end
26     return O;
1  Procedure sampleAndAverage(T,a)
2      i ← 0
3      inst ← newZeroVector
4      while i ≤ a do
5          x ← randomSample(T)
6          inst ← inst + (x/a)
7          i ← i + 1
8      end
9      return inst;
```

domly sampled tweets from *posT* or *posN* have the desired polarity. This is equivalent to saying that at least $\lfloor \frac{a}{2} \rfloor + 1$ tweets from the sample have the desired positive or negative polarity. If we assume that the tweets in *posT* and *negT* are independent and identically distributed (IID), the probability of M can be calculated by adding the values of the Binomial probability mass function from $\lfloor \frac{a}{2} \rfloor + 1$ to a. This corresponds to adding all the cases in which more

than the half of the sampled tweets (the majority) have the desired polarity. This probability is calculated as follows:

$$P(M) = \sum_{i=\lfloor \frac{a}{2} \rfloor + 1}^{a} \binom{a}{i} p_d^i (1 - p_d)^{a-i}$$

Note that this value is equivalent to 1 minus the cumulative distribution function of the Binomial distribution evaluated at $\lfloor \frac{a}{2} \rfloor$. The probabilities of M for different values of a ($a \geq 3$) and p_d ($p_d > 0.5$) are shown in Table 6.1.

From the table, we observe that all the calculated probabilities are greater than p_d and generally increase when increasing p_d or a (exceptions occur when switching from an odd to an even number of votes). Thus, assuming the lexical polarity hypothesis is true and thus $p_d > 0.5$ for *posT* and *negT*, we can say that the majority of the tweets sampled by ASA have the desired polarity with a probability greater than p_d. Moreover, we can expect that the instances produced by ASA will behave similarly to the majority of the instances they are obtained from. Thus, compared to sampling individual tweets, we can have greater confidence that ASA instances will be in the desired polarity region of the attribute space.

The ideas discussed above are inspired by Condorcet's Jury Theorem, which is used in the context of decision making. The theorem states that if a random individual votes for the correct decision with probability $p_d > 0.5$, the probability of the majority being correct tends to one when increasing the number of independent voters. This is a consequence of the law of large numbers, and as was shown in (Ladha, 1993), the same conclusions can be obtained after relaxing the independence assumption.

In our problem, each tweet sampled from *posT* or *negT* can be interpreted as a vote for the polarity of the averaged instance. We expect a trade-off in the value of a. While a small value of a will decrease the confidence of generating

	$p_d = 0.6$	$p_d = 0.7$	$p_d = 0.8$	$p_d = 0.9$
$a = 3$	0.648	0.784	0.896	0.972
$a = 5$	0.683	0.837	0.942	0.991
$a = 10$	0.633	0.850	0.967	0.998
$a = 50$	0.902	0.998	1	1
$a = 100$	0.973	1	1	1
$a = 500$	1	1	1	1
$a = 1000$	1	1	1	1

Table 6.1: Probabilities of sampling a majority of tweets with the desired polarity.

an instance with the target polarity, a very large value will generate instances that, despite being likely to have the right label, will be very similar to each other. This could affect the generalisation ability of the tweet classifier trained from those instances.

The resulting training dataset $\mathcal{O}$ is created by repeating the sampling and averaging steps p times for the positive class and n times for the negative one. The pseudo-code of ASA is given in Algorithm 6.1.1.

Setting the flag m in the algorithm will generate polarity instances from tweets in which words from the opposite polarity are never observed. Considering that positive and negative tweets are likely to contain words with the opposite polarity, we expect that unsetting the flag will produce instances with better generalisation properties. Both setups are compared in Section 6.4.

We use ASA for creating balanced training data by setting p and n to the same value. This is done to address the sentiment imbalance problem discussed in (Li et al., 2011): classifiers trained from imbalanced datasets may have difficulties recognising the minority class. The balancing properties of ASA are inspired by a well-known resampling technique used for training classifiers from imbalanced datasets called Synthetic Minority Over-sampling Technique (SMOTE) (Chawla, Bowyer, Hall and Kegelmeyer, 2002). SMOTE oversamples the minority class by generating synthetic examples for the minority class. Each new instance is calculated as a random weighted average between an existing example of the minority class and one of its nearest neighbours. The similarity between ASA and SMOTE is that both methods generate new instances by averaging existing ones. The difference is that in ASA the average is unweighted and can involve more than two examples. Furthermore, ASA does not require calculating the distance between the examples being averaged. This is a convenient aspect of ASA considering that tweets are represented by high-dimensional vectors. Another important difference relates to the type of data used for generating the instances. SMOTE combines labelled instances; ASA combines unlabelled instances annotated using an opinion lexicon.

6.2 ASA and The Tweet Centroid Model

The tweet centroid model is a generic framework that has been used in this thesis for three different tasks: 1) polarity lexicon induction from a seed lexicon (Chapter 4), 2) polarity lexicon induction from tweets annotated by sentiment (Chapter 5), and 3) classifying the sentiment of tweets using word vec-

tors labelled by a polarity lexicon (Chapter 5). In contrast, ASA is purely a lexicon-based distant supervision model.

ASA and the partitioned version of the tweet centroid model (when used as a lexicon-based distant supervision method), share a very important characteristic: they both generate labelled instances by averaging multiple tweet vectors annotated with a given polarity lexicon.

A difference between the two models is the type of tweets they average. The training instances produced by the partitioned tweet centroid model are word vectors created by averaging tweet vectors containing the same word. ASA, on the other hand, averages tweets from sets *posT* and *negT* that do not necessarily have any word in common.

In terms of implementation, the tweet centroid model requires an inverted index to map all words from the vocabulary to the tweets in which they occur. ASA is cheaper to implement since it only maps the two sets *posT* and *negT* to the corresponding tweets with positive and negative words respectively.

The models also differ in the size of the training data they generate. When the partition size of the partitioned tweet centroid model is set to a small number (as suggested by the results shown in Table 5.2) and the source corpus is large, the number of generated training instances can be large. This could be a problem for training models that do not scale well to large training datasets due to time or memory constrains. ASA on the other hand, allows unconstrained specification of the number of generated positive and negative instances. This can be useful for building compact training datasets.

The tweet centroid model ensures that all tweets annotated with the polarity lexicon are used at least once for building the training instances. ASA is likely to discard valuable information because it randomly samples tweets from the sets *posT* and *negT*.

Another difference is the label distribution of the datasets the two methods generate. As positive words occur more frequently than negative ones (Table 5.1), the partitioned tweet centroid model tends to create unbalanced training datasets. ASA can easily deal with this problem because it can be parametrised to generate the same amount of positive and negative instances.

We will observe in Section 6.4, that despite these differences, ASA and the tweet centroid model produce similar classification performance on tweets after tuning their corresponding parameters.

6.3 The Lexical Polarity Hypothesis

The word-tweet sentiment-interdependence relation studied in Chapter 5 tells us that the sentiment of words and tweets are strongly interrelated. In this section, we study the hypothesis called "lexical polarity hypothesis" on which ASA is based. It extends the second part of the word-tweet sentiment-interdependence relation: the sentiment of a tweet is determined by the sentiment of its words.

The lexical polarity hypothesis encapsulates the idea that, since tweets are short messages, the presence of a single opinion word is a very strong indicator of the polarity of the message. The hypothesis is expressed in the following two statements:

1. A tweet containing at least one positive word is more likely to be positive than negative.

2. A tweet containing at least one negative word is more likely to be negative than positive.

We study this hypothesis empirically by estimating the probabilities of events corresponding to these statements using the *SemEval* corpus of hand-annotated positive and negative tweets and the AFINN lexicon. We take a balanced sample of 2000 positive and 2000 negative tweets from *SemEval* to avoid bias caused by unevenly distributed tweets and focus the analysis on how the polarity of tweets is affected by the polarity of their words. Hence, we calculate the sets *posT* and *negT* from this corpus and study the polarity distribution of their messages.

We first study the distribution of *posT* and *negT* by unsetting the m flag. Hence, we include tweets with mixed positive and negative words in both sets. The set *postT* has 2419 tweets, which corresponds to 60% of the tweets, and has a distribution of 826 negative and 1593 positive tweets. Thus, the estimated probability of a tweet from *posT* having a positive polarity is 0.66. The set *negT* contains 1774 tweets, corresponding to 44% of the tweets, and has a distribution of 1354 negative and 420 positive tweets. This gives an estimated probability of 0.76 that a tweet from *negT* is negative. These results suggest that negative words are stronger indicators than positive words for determining the polarity of a tweet.

We also study the distribution of *posT* and *negT* after discarding tweets with mixed positive and negative words (m turned on). In this case, the size of *posT* is reduced to 1552 (39% of the total) tweets with a distribution of 284 negative

and 1268 positive tweets. This gives an estimated probability of 0.817 that a tweet from *posT* is positive. The size of *negT* is reduced to 907 tweets (23% of the total) with a distribution of 812 negative and 95 positive tweets. This gives an estimated probability of 0.9 that a tweet from *negT* is negative.

The polarity distributions of these sets are presented as bar charts in Figure 6.1. The figure shows how the distributions become more skewed when removing tweets with mixed positive and negative opinion words.

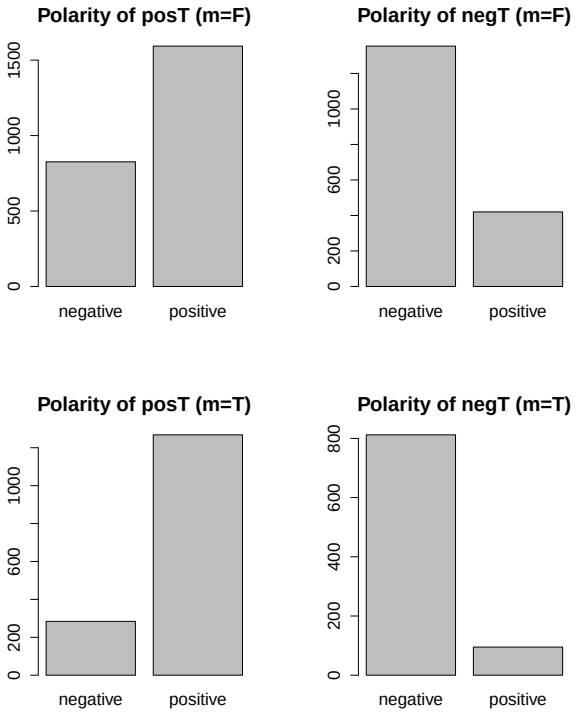

Figure 6.1: Polarity distributions of *posT* and *negT*.

We also study the distribution of tweets with mixed positive and negative words. We found 857 such tweets (21% of the total) with a distribution of 542 negative and 325 positive tweets. These numbers also indicate that negative opinion words have a greater effect than positive words on the polarity of the tweets in which they occur. However, negative words are also less frequent than positive ones.

The results obtained in this section support the lexical polarity hypothesis on which ASA is based. We can conclude that opinion words are indeed strong indicators of the polarity of tweets. We observed that discarding tweets with mixed opinion words produces a stronger effect. However, it is important to bear in mind that discarding these tweets may also cause loss of valuable in-

formation. The effects of averaging multiple tweets containing opinion words with the same polarity are investigated in the following section.

6.4 Classification Experiments

In this section, we conduct an experimental evaluation of the proposed ASA algorithm. We evaluate the performance of classifiers trained with instances generated by ASA on the same three datasets of hand-annotated tweets that were used in the other chapters of this thesis: *6HumanCoded*, *Sanders*, and *SemEval*.

We consider the same distant supervision baselines used in Chapter 5: the emoticon-annotation approach (EAA) and the lexicon-annotation approach (LAA). It is interesting to observe that the positive and negative instances from LAA are equivalent to the sets *posT* and *negT* from ASA when m is turned on.

Considering that positive signals such as positive emoticons or positive opinion words occur more frequently in tweets than their negative counterparts, we also study balanced versions of EAA and LAA. The balanced baselines are referred to as EAA_B and LAA_B, and are obtained by undersampling the majority class in each case.

We again take our unlabelled tweets from the Edinburgh corpus (ED) and use unigrams, POS tags, and Brown word clusters for representing the tweets in a feature space.

With the aim of analysing the effect of averaging multiple tweets for building training instances, we study different values of the a parameter of ASA. We also study the effect of including or excluding tweets with mixed positive and negative words by comparing the performance of ASA with the flag m turned on and off respectively. We create balanced and compact training datasets with size equal to 1% of the size of the source corpus by setting the parameters p and n to 0.5% of the source corpus size, corresponding to a subset of the Edinburgh corpus (see below).

The remainder of the experimental setup is analogous to the one of Chapter 5 for transferring sentiment knowledge from words to tweets. The classifiers are trained using the same logistic regression algorithm taken from LIBLINEAR, with the regularisation parameter C set to 1.0, and the tweets from the target collections are mapped into the same feature space as the tweets generated by the distant supervision models. We train each distant supervision ten times on data generated from the same ten independent partitions of 2 million tweets from the Edinburgh corpus. The average performance of

each classifier trained with ASA is compared with the average performance of classifiers trained with each of the four distant supervision baselines 1) EAA, 2) EAA_B, 3) LAA, 4) LAA_B, using a paired Wilcoxon signed-rank test with the significance value set to 0.05.

As was already shown in Table 5.1, different distant supervision models produce different numbers of labelled instances from the same corpus of unlabelled tweets. The average number of positive and negative instances generated by all the distant supervision schemas evaluated in this chapter are shown in Table 6.2.

We use the macro-averaged F1 score in addition to the AUC measure used in Chapter 5 as evaluation criteria. Macro-averaged F1 was used in the SemEval Twitter sentiment analysis task[3].

The comparisons are done for each target collection of tweets and the results for the macro-averaged F1 score and AUC are given in Table 6.3. The statistical significance tests of each configuration of ASA with respect to each of the four baselines are indicated by a sequence of four symbols. Improvements are denoted by a plus (+), degradations by a minus (-), and cases where no statistical significant difference is observed by an equals (=). The baselines are also compared amongst each other.

We observe that EAA performs substantially worse than the other baselines in F1 score. EAA_B performs substantially better than EAA. From Table 6.2 we observe that EAA is the model that produces the most uneven distribution of positive and negative instance. This suggest that the macro-average F1 score is very sensitive to classifiers trained from heavily imbalanced data. In contrast, we can note that balancing EAA does not cause any improvement in AUC. AUC is a more robust measure for classifiers trained from imbalanced datasets.

Regarding the LAA baseline, we observe a degradation in F1 after balancing

	Avg. Positive	(%)	Avg. Negative	(%)	Avg. Total	(%)
EAA	$130,641$	(6.5%)	$21,537$	(1.1%)	$152,179$	(7.6%)
EAA_B	$21,537$	(1.1%)	$21,537$	(1.1%)	$43,074$	(2.2%)
LAA	$681,531$	(34.1%)	$294,177$	(14.7%)	$975,708$	(48.8%)
LAA_B	$294,177$	(14.7%)	$294,177$	(14.7%)	$588,354$	(29.4%)
ASA	$10,000$	(0.5%)	$10,000$	(0.5%)	$20,000$	(1%)

Table 6.2: Average number of positive and negative instances generated by different distant supervision models from 10 collections of 2 million tweets.

[3]http://alt.qcri.org/semeval2016/task4/

Macro-averaged F1										
	6HumanCoded				Sanders				SemEval			
EAA_U	0.576 ± 0.007	=	-	-	-	0.506 ± 0.018	=	-	-	-	0.591 ± 0.018	=	-	-	-
EAA_B	0.735 ± 0.008	+	=	+	+	0.709 ± 0.018	+	=	=	=	0.711 ± 0.006	+	=	-	=
LAA_U	0.729 ± 0.004	+	-	=	+	0.711 ± 0.003	+	=	=	+	0.725 ± 0.002	+	+	=	+
LAA_B	0.719 ± 0.002	+	-	-	=	0.703 ± 0.004	+	=	-	=	0.712 ± 0.002	+	=	-	=
ASA ($a = 1$, $m = T$)	0.734 ± 0.005	+	=	+	+	0.721 ± 0.010	+	+	+	+	0.724 ± 0.004	+	+	=	+
ASA ($a = 5$, $m = T$)	0.745 ± 0.005	+	+	+	+	0.723 ± 0.010	+	+	+	+	0.722 ± 0.006	+	+	=	+
ASA ($a = 10$, $m = T$)	0.737 ± 0.003	+	=	+	+	0.703 ± 0.011	+	=	-	=	0.708 ± 0.007	+	-	-	=
ASA ($a = 50$, $m = T$)	0.693 ± 0.003	+	-	-	-	0.643 ± 0.004	+	-	-	-	0.639 ± 0.006	+	-	-	-
ASA ($a = 100$, $m = T$)	0.672 ± 0.004	+	-	-	-	0.620 ± 0.005	+	-	-	-	0.607 ± 0.006	+	-	-	-
ASA ($a = 500$, $m = T$)	0.638 ± 0.004	+	-	-	-	0.599 ± 0.008	+	-	-	-	0.563 ± 0.005	-	-	-	-
ASA ($a = 1000$, $m = T$)	0.635 ± 0.004	+	-	-	-	0.594 ± 0.010	+	-	-	-	0.554 ± 0.003	-	-	-	-
ASA ($a = 1$, $m = F$)	0.717 ± 0.007	+	-	-	=	0.691 ± 0.013	+	-	-	-	0.699 ± 0.008	+	-	-	-
ASA ($a = 5$, $m = F$)	0.755 ± 0.004	+	+	+	+	0.730 ± 0.008	+	+	+	+	0.735 ± 0.005	+	+	+	+
ASA ($a = 10$, $m = F$)	**0.761** ± 0.003	+	+	+	+	**0.735** ± 0.015	+	+	+	+	**0.742** ± 0.006	+	+	+	+
ASA ($a = 50$, $m = F$)	0.749 ± 0.004	+	+	+	+	0.673 ± 0.005	+	-	-	-	0.699 ± 0.009	+	-	-	-
ASA ($a = 100$, $m = F$)	0.717 ± 0.003	+	-	-	-	0.645 ± 0.006	+	-	-	-	0.664 ± 0.005	+	-	-	-
ASA ($a = 500$, $m = F$)	0.665 ± 0.002	+	-	-	-	0.621 ± 0.007	+	-	-	-	0.621 ± 0.004	+	-	-	-
ASA ($a = 1000$, $m = F$)	0.653 ± 0.003	+	-	-	-	0.619 ± 0.007	+	-	-	-	0.613 ± 0.002	+	-	-	-
AUC										
	6HumanCoded				Sanders				SemEval			
EAA_U	0.805 ± 0.005	=	=	-	-	0.800 ± 0.017	=	=	+	+	0.802 ± 0.006	=	+	-	-
EAA_B	0.809 ± 0.001	=	=	=	=	0.795 ± 0.016	=	=	+	+	0.798 ± 0.007	-	=	-	-
LAA_U	0.809 ± 0.001	+	=	=	=	0.778 ± 0.002	-	-	=	=	0.814 ± 0.000	+	+	=	=
LAA_B	0.809 ± 0.001	+	=	=	=	0.778 ± 0.003	-	-	=	=	0.813 ± 0.001	+	+	=	=
ASA ($a = 1$, $m = T$)	0.806 ± 0.003	=	=	-	-	0.786 ± 0.007	-	-	+	+	0.808 ± 0.002	+	+	-	-
ASA ($a = 5$, $m = T$)	0.809 ± 0.002	=	=	=	=	0.787 ± 0.005	-	=	+	+	0.810 ± 0.003	+	+	-	-
ASA ($a = 10$, $m = T$)	0.804 ± 0.001	=	-	-	-	0.776 ± 0.008	-	-	=	=	0.806 ± 0.003	+	+	-	-
ASA ($a = 50$, $m = T$)	0.756 ± 0.003	-	-	-	-	0.697 ± 0.005	-	-	-	-	0.763 ± 0.002	-	-	-	-
ASA ($a = 100$, $m = T$)	0.729 ± 0.002	-	-	-	-	0.672 ± 0.006	-	-	-	-	0.739 ± 0.002	-	-	-	-
ASA ($a = 500$, $m = T$)	0.696 ± 0.003	-	-	-	-	0.642 ± 0.008	-	-	-	-	0.707 ± 0.005	-	-	-	-
ASA ($a = 1000$, $m = T$)	0.690 ± 0.004	-	-	-	-	0.637 ± 0.008	-	-	-	-	0.701 ± 0.006	-	-	-	-
ASA ($a = 1$, $m = F$)	0.793 ± 0.005	-	-	-	-	0.762 ± 0.016	-	-	-	-	0.787 ± 0.007	-	-	-	-
ASA ($a = 5$, $m = F$)	0.837 ± 0.004	+	+	+	+	0.807 ± 0.010	=	=	+	+	0.833 ± 0.003	+	+	+	+
ASA ($a = 10$, $m = F$)	**0.845** ± 0.001	+	+	+	+	**0.812** ± 0.015	+	+	+	+	**0.840** ± 0.003	+	+	+	+
ASA ($a = 50$, $m = F$)	0.815 ± 0.003	+	+	+	+	0.759 ± 0.006	-	-	-	-	0.810 ± 0.004	+	+	-	-
ASA ($a = 100$, $m = F$)	0.781 ± 0.003	-	-	-	-	0.720 ± 0.007	-	-	-	-	0.779 ± 0.004	-	-	-	-
ASA ($a = 500$, $m = F$)	0.723 ± 0.002	-	-	-	-	0.670 ± 0.008	-	-	-	-	0.729 ± 0.005	-	-	-	-
ASA ($a = 1000$, $m = F$)	0.712 ± 0.002	-	-	-	-	0.665 ± 0.007	-	-	-	-	0.721 ± 0.005	-	-	-	-

Table 6.3: Macro-averaged F1 and AUC measures for different distant supervision models. Best results per column for each measure are given in bold.

the data (LAA_B). On the other hand, LAA_B performs almost identically to LAA in AUC. We believe that the reason why balancing is not causing a positive impact in the lexicon-based approach is that LAA produces a less skewed distribution of positive and negative instances than EAA. The benefits of resampling are more substantial for F1 for very skewed distributions such as those produced by EAA.

As was already pointed out in Chapter 5, there is no clear consensus about which baseline is the best. The baselines based on lexicons perform better

than the ones based on emoticons when evaluating on SemEval, for both F1 and AUC. On Sanders, the lexicon and the balanced emoticons behave similarly in F1, but the emoticons perform better for AUC. On 6HumanCoded, EAA_B performs better than LAA and LAA_B in F1, but in AUC they produce almost identical results. It is worth mentioning that the emoticon-based approach can achieve competitive results to the lexicon-based one even though it generates substantially less training data (Table 6.2).

Regarding ASA, we observe that the performance achieved by our proposed method depends on the parameter setting (Table 6.3). When tweets with mixed positive and negative tweets are discarded (m=T) we observe that the best results are achieved when very few tweets are averaged. There is a strong decline in the performance of ASA (m=T) when the value of a is increased. We believe that this is because instances become too similar when formed by averaging too many tweets. ASA (m=T) with a=1 is essentially a subsampled version of LAA_B, and indeed produces very similar results. ASA (m=T) is not capable of producing statistically significant improvements over the four baselines for either AUC and F1 score for any dataset, even when considering the optimum value of a. This suggests that there is no clear contribution in the sample and average steps of ASA when tweets with mixed positive and negative tweets are discarded.

On the oher hand, when tweets with mixed positive and negative words are simultaneously added to both sets (m=F), ASA produces statistically significant improvements over all the baselines in all target collections for both F1 and AUC, for appropriate values of a. The best value of a is ten in all three target collections, for both performance metrics. These results indicate that ASA, with calibrated parameters, outperforms existing distant supervision models for Twitter polarity classification. The fact that turning m off is better than discarding tweets with mixed positive and negative words suggests that mixed tweets contribute to better generalisation. This is because real positive and negative tweets are likely to contain words with both polarities.

We clearly observe that setting a to one in ASA (m=F) produces results that are far from the optimum. This validates the idea that averaging multiple tweets with at least one word of the same polarity increases the chance of producing an instance of the desired polarity. We observe again a decline in performance when the value of a is increased beyond ten.

Based on the numbers in Table 6.2 we use 7.6 and 48.8 times more training data with EAA and LAA than with ASA respectively. It is noteworthy that ASA

classifiers outperform the classifiers trained with EAA and LAA even though they are trained with less data. This shows that ASA can produce a more compact and efficient training dataset than previous distant supervision models.

Moreover, if we compare the AUC values of ASA and TCM from Chapter 5 (when the latter is used as a distant supervision model), after tuning the corresponding parameters a and m for ASA and p for TCM (Table 5.2), we observe that both models are very competitive. Although ASA exhibits a slightly worse performance, TCM with p set to 20 generates around 5 times more instances than ASA. This result provides further support that ASA can produce effective and compact training datasets for Twitter polarity classification.

6.4.1 Sensitivity Analysis

In the previous experiments, the number of instances generated by ASA was fixed to create balanced datasets whose size was 1% of the size of the source corpus. In the next experiment, we explore the performance of ASA when manipulating the number of instances generated for both settings of m and using different values of a. We use values of a smaller than 20 because large values of this parameter produced very poor results in the previous experiment. We train a grid of polarity classifiers on instances generated by ASA from a corpus of 2 million tweets, which are then deployed on the SemEval dataset. The number of generated instances ranges from 10 thousand to 400 thousand, and the value of a ranges from 1 to 20. The macro-averaged F1 and AUC scores obtained by classifiers trained with ASA instances using different values of a, m, and the number of generated instances, are shown in the heatmaps of Figure 6.2.

The darker a cell in the heatmap, the higher the performance achieved by the corresponding configuration of parameters. We observe again that setting m to false produces higher performance (darker cells) than setting m to true.

In relation to the cells for m=T, we observe that the highest F1 values (darkest cell) are when a is equal to 5. We also observe in this setting that most of the cells are very similar to each other for AUC. Generating less than 20 thousand instances produces brighter cells in several cases in this setting, and we do not observe substantial improvements when increasing the number of generated instances beyond 20 thousand.

We observe that setting a to 1 produces poorer results (brighter cells) for m=F. In general, we observe that turning m off and generating more instances improves the performance of the classifier for both F1 and AUC.

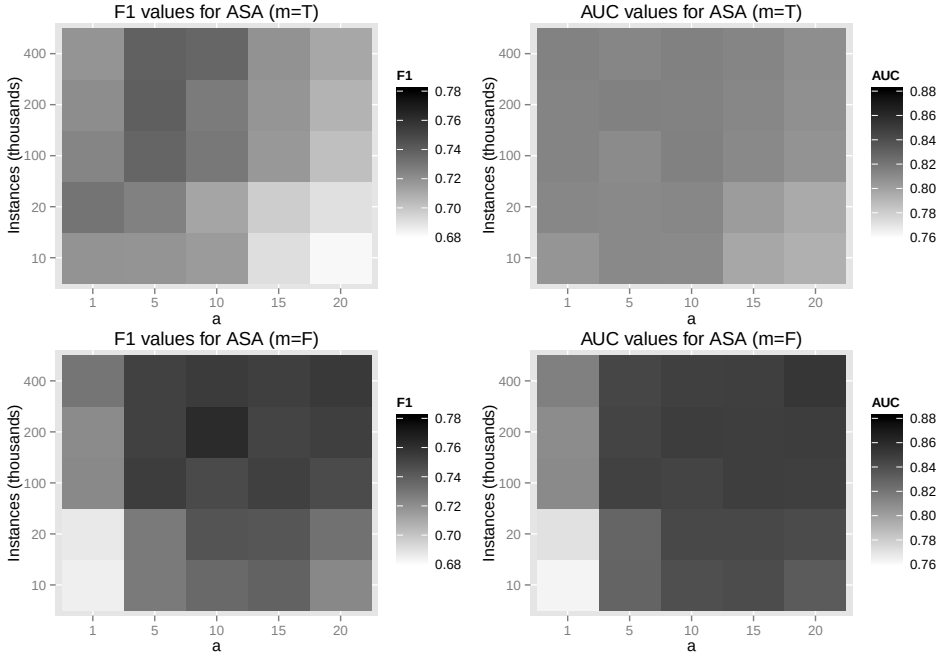

Figure 6.2: Heatmap of ASA parameters on the SemEval dataset. The highest F1 value for m=F is 0.76 ($a = 10$, instances = 200), and for m=T is 0.74 ($a = 5$, instances = 20). The highest AUC values for m=F and m=T occur with the same configurations as the highest values for F1 and are 0.85 and 0.81, respectively.

However, the best results are not necessarily obtained when the maximum number of instances is generated (400 thousand). For example, the best F1 score is achieved with 200 thousand instances.

6.4.2 Learning Curves

We also study the effect of increasing the source corpus size in all different distant supervision methods: EAA, EAA_B, LAA, LAA_B, and ASA. It is important to remark that the number of generated instances in the four distant supervision baselines increases when increasing the size of the source corpus. The increments are proportional to the percentages shown in Table 6.2.

We trained classifiers using partitions of the source corpus ranging from ten thousand to ten million tweets. For the ASA model we set a to 10 and m to false, which were the best parameters according to the previous experiments (Table 6.3), and kept p and n with values set to $0.005 \times |\mathcal{C}|$, for generating balanced datasets with size equal to 1% of the size of the source corpus. Thus, the number of generated instances in ASA is also increased when using a larger source corpus.

The learning curves produced by logistic regressions applied to the SemEval dataset, trained with data generated using ASA and the four baselines from source corpora of different sizes, are shown in Figure 6.3. The performance metrics are again the macro-averaged F1 measure and AUC.

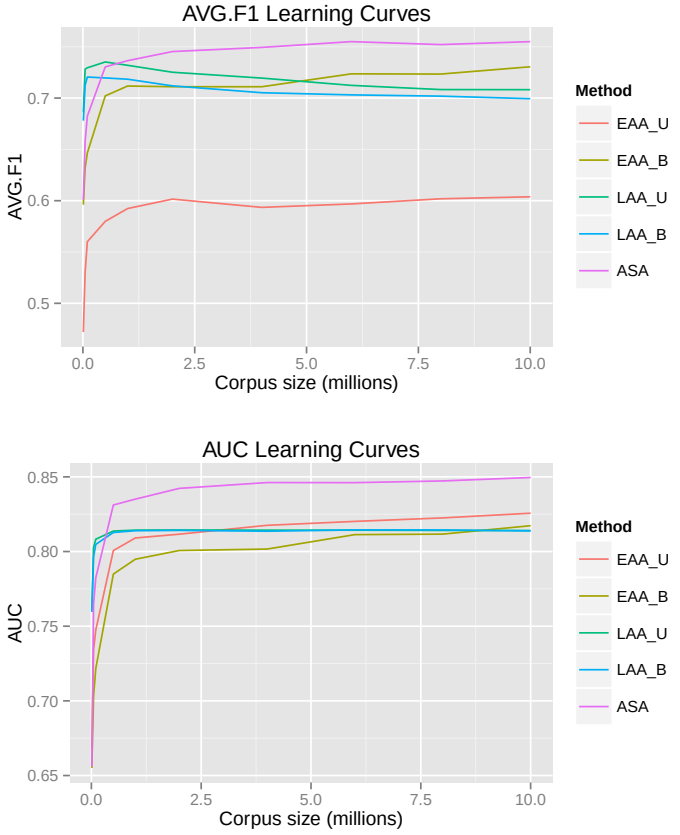

Figure 6.3: Learning curves for the SemEval dataset.

The figure indicates that most methods increase their performance when increasing the corpus size, and that these improvements tend to plateau when using more than 2 million tweets as input. We observe again that EAA exhibits poor performance in F1 and that balancing this method (EAA_B) produces substantial improvements for this measure. Surprisingly, the lexicon-based baselines LAA and LAA_B exhibit a slight decrease in F1 when increasing the source corpus size after the million tweet mark.

We observe in the initial part of the curves that LAA and LAA_B are the best distant supervision methods for source corpora smaller than 1 million tweets. This suggests that the prior knowledge from the lexicon can be very useful with small collections of data. It is important to note that the setup of ASA for this experiment generates very few examples when the source corpus is small.

This can be easily changed by generating more training data when the source corpus is too small.

We also observe that after passing the million tweet mark, the emoticon-based models are better than LAA and LAA_B, and that ASA outperforms all the other models. These results indicate that ASA is a powerful distant supervision model that can be used for training accurate message-level polarity classifiers without relying on very large collections of unlabelled data.

6.4.3 Qualitative Analysis

In this subsection we explore some tweets classified by ASA. Examples of tweets from the SemEval corpus classified using ASA, with $a = 10$ and $m = F$, are given in Table 6.4. The positive and negative words from the AFINN lexicon in these tweets are marked with blue and red colours respectively.

	Positive Tweets
f(x)=neg	Never start working on your **dreams** and goals tomorrow... tomorrow never comes.... if it means anything to U, ACT NOW! #getafterit
	Just did Spartacus 2.0 and sauna. . . imma be **sore** tomorrow but so **worth** it
	@patrishuhx7 I have English tomorrow but it honestly doesn't **bother** me for some reason. Rella always makes my day. Don't ask
f(x)=pos	**Happy** Valentine's Day!!! @MAziing: Everyday is the 14th!
	Ground hog day is such a **good** film, Sunday is for food and films #sunday
	Going to see Kendrick Lamar with @Pea_Starks in jan :D
	Negative Tweets
f(x)=neg	Can we just haw class **cancelled** tomorrow? Cause I really don't **want** to go to BCA 101. I'd rather eat worms....
	I never had a **good** time, I sat by my bedside. With papers and poetry about Estella
	I got tickets to the NC State game saturday and nobody to go with..
f(x)=pos	**Wish** me **lucky** on the Cahsee tomorrow I'm **pretty** **nervous**
	I haven't talked to you since July 19 th and all you can say is So do you **like** Beyonce's new cd GTFO
	Being in Amsterdam this early on a friday morning is not my ideal, I just **want** to get home!

Table 6.4: Examples of tweets classified with ASA. Positive and negative words from AFINN are marked with blue and red colours respectively. The leftmost column indicates the classifier's output.

The classification outputs reveal some insights about the strengths and shortcomings of our method. The correctly classified examples suggest that ASA is capable of learning sentiment expressions that go beyond the lexicon used in the annotation phase. This is observed in the second and third negative examples, and the last positive one, which are all correctly classified even though they do not contain AFINN words with the same polarity than the corresponding tweet. ASA learns opinion words co-occurring with the words from the lexicon, because all words from a tweet are considered in the feature space. This is an indirect form of polarity lexicon expansion. Regarding the misclassified examples, we observe that the current implementation of ASA is not capable of accurately handling complex sentiment patterns involving negations and *but* clauses. We attribute these problems to two factors: 1) the annotation phase is solely based on unigrams, and 2) the current feature space omits the order

in which words occur. The first factor could be addressed by using a lexicon of sentiment annotated phrases, and the second one by using more sophisticated feature representations such as n-grams or paragraph vector-embeddings (Le and Mikolov, 2014).

6.5 Discussion

We propose a new model called ASA to generate synthetic training data for Twitter sentiment analysis from unlabelled corpora using the prior knowledge provided by an opinion lexicon[4]. The method annotates tweets according to the polarity of their words, using a given polarity lexicon, and generates balanced training data by sampling and averaging tweets containing words with the same polarity. ASA is based on the lexical polarity hypothesis: because tweets are short messages, opinion words are strong indicators of the sentiment of the tweets in which they occur, and therefore tweets with at least one word with a certain known prior polarity are more likely to express the same polarity on the message level than the opposite one. The sample and average steps of ASA exploit this hypothesis by increasing the confidence of generating an instance located in the desired polarity region. ASA also incorporates a novel way for incorporating the knowledge provided by tweets with mixed positive and negative words.

The experimental results show that ASA produces better classifiers than the widely-adopted approach of using emoticons for labelling tweets into polarity classes and also better results than labelling tweets based on the polarity of their words, without sampling and averaging. Moreover, classifiers trained with data generated by ASA achieve competitive results when compared to the partitioned tweet centroid model presented in Chapter 5, using substantially less training data. This shows that ASA can generate compact and efficient datasets for learning polarity concepts.

In the same way as the partitioned tweet centroid model, ASA can be used for training Twitter polarity classifiers in scenarios without labelled training data and for creating domain-specific sentiment classifiers by collecting unlabelled tweets from the target domain.

[4]The source code of the model is available for download at http://www.cs.waikato.ac.nz/ml/sa/ds.html#asa.

Chapter 7

Conclusions

This thesis addresses the problem of classifying tweets into sentiment classes when labels are scarce. We developed polarity lexicon induction and distant supervision models aimed at answering the research question stated in Chapter 1:

> "Polarity classification of tweets when training data is sparse can be successfully tackled through Twitter-specific polarity lexicons and lexicon-based distant supervision."

All the models proposed in this thesis either attempt to acquire new lexical knowledge or exploit existing lexical knowledge for analysing the sentiment of tweets. The word-sentiment association model and the tweet centroid model, described in Chapters 3 and 4 respectively, focus on the acquisition of lexical knowledge by automatically inducing Twitter-specific opinion lexicons. The partitioned version of the tweet centroid model (Chapter 5) and the ASA algorithm (Chapter 6) both exploit lexical knowledge to generate annotated data for training polarity classifiers from unlabelled tweets.

This chapter provides a summary of the main findings. The chapter is structured as follows. A summary of the results is provided in Section 7.1. The main contributions of this thesis are given in Section 7.2. Section 7.3 points out some possibilites for future work.

7.1 Summary of Results

The main results of this thesis can be stated as follows:

- The proposed word-sentiment association method for polarity lexicon induction studied in Chapter 3 improves on the three-dimensional word-level polarity classification performance obtained by using PMI-SO alone.

This is significant because PMI-SO is a state-of-the-art measure for establishing world-level sentiment.

- As shown in Chapter 3, POS tags are effective features for discriminating between neutral and non-neutral words.
- The lexicons created with the word-sentiment association method presented in Chapter 3 achieve significant improvements over SentiWordNet when classifying tweets into polarity classes, and also outperform SentiStrength in most of the experiments. This is significant because SentiWordNet and SentiStrength are well known resources for sentiment analysis.
- The tweet centroid model is a better weighting scheme than positive PMI for building distributional word vectors in the polarity lexicon induction task (Chapter 4).
- The word-level polarity classification experiments for the tweet centroid model (Chapter 4) show that representations built from unigram features and Brown clusters complement each other in a statistically significant manner.
- The lexicons induced with the word-sentiment association method and the tweet centroid model produce significant improvements over the seed lexicon for tweet-level polarity classification (Chapters 3 and 4).
- Low-dimensional word-embeddings are better than distributional word-level features obtained by averaging tweet-level features when performing multi-label classification of Twitter words into emotions (Chapter 4).
- The results obtained in Chapter 5 show that the proposed tweet centroid model is better than PMI-SO for classifying the sentiment of words from sentiment-annotated tweets.
- The experimental results of Chapter 5 and 6 show that the training datasets generated by partitioned tweet centroids and ASA (after tuning their corresponding parameters) produce classifiers that perform significantly better than a classifier trained from emoticon-annotated tweets and a classifier trained from tweets annotated according to the polarity of their words. Furthermore, ASA achieves a similar performance than partitioned tweet centroids using less training data.

The results listed above suggest that Twitter-specific polarity lexicons and lexicon-based distant supervision methods can successfully tackle the polarity classification of tweets when labels are scarce. Therefore, we can conclude that our research hypothesis is supported by the experimental results.

7.2 Contributions

The main research contributions of this thesis are listed below:

- The finding that two types of word associations calculated from sentiment-annotated tweets, PMI and SGD, can be effectively combined with POS tags for classifying words into positive, negative, and neutral sentiment categories (Chapter 3).
- The proposal of a new version of PMI-SO that can be calculated from soft-annotated tweets (Chapter 3).
- A new window-free distributional model for building word vectors from tweets: the tweet-centroid model. This model can be used together with any message-level feature representation and was shown to work better than positive PMI for polarity lexicon induction (Chapter 4).
- A new framework for determining word-emotion associations based on distributional word vectors and multi-label classification that, in contrast to previous work, does not depend on tweets annotated with emotional hashtags (Chapter 4).
- A new framework for transferring sentiment knowledge between words and tweets based on representing them by feature vectors of the same dimensionality (Chapter 5).
- A new distant supervision approach that builds balanced and compact training datasets for message-level polarity classification and outperforms the well-known emoticon-annotation approach (Chapter 6).

In addition to these contributions, there are two reasons that make us believe that the models developed in this thesis offer a practical framework for people who are not necessarily part of the NLP community (e.g, journalists, sociologists) to analyse public opinion from tweets:

1. The source code of all the models is freely available for download and is integrated into the well known *Weka* machine learning software.

2. None of the models depend on tweets that were manually annotated by sentiment. This allows users to induce polarity lexicons or train message-level polarity classifiers from any collection of tweets without incurring the cost of annotating tweets by sentiment.

7.3 Future Work

The results obtained in this thesis open several directions for further research. In this section we discuss possible extensions to the different models proposed in this thesis.

7.3.1 Extensions to the Word-Sentiment Association Method

Our supervised framework for lexicon expansion based on word-sentiment associations and POS tags can be extended in multiple ways. For instance, this approach could be used for creating a concept-level opinion resource for Twitter by employing word clustering techniques such as the Brown Clustering method (Brown et al., 1992). We could build the same time series we have built for words for word clusters, and use the trained classifier for estimating a sentiment distribution for each word cluster or concept.

We believe that unlabelled words and their feature values could provide valuable information that is not being exploited so far. Semi-supervised methods such as the EM algorithm (Nigam, McCallum and Mitchell, 2006) could be used to include unlabelled words as part of the training process.

Because our word-level features are based on time series, they could be easily calculated in an on-line fashion from a stream of time-evolving tweets. Based on this, we could study the dynamics of opinion words. New opinion words could be discovered because the change of the distribution in certain words could be tracked.

7.3.2 Extensions to the Tweet Centroid Model

The current version of the tweet centroid model for polarity lexicon induction creates large dimensional vectors. Considering that low dimensional dense vectors trained with *word2vec* embeddings (Mikolov et al., 2013) produce better results than the tweet centroid model when performing multi-label classification of words into emotions (Chapter 4), we would like to explore low-dimensional projections of the tweet centroid model obtained by training auto-encoders or restricted Boltzman machines on this representation.

In a similar way as was proposed for the word-sentiment association method, we could create time-evolving word-sentiment associations from Twitter streams using an incremental version of the tweet centroid model. The incremental version would require updating the counts of the word vectors when new tweets arrive, and also involve creating new vectors when new words are found. An efficient way for maintaining these vectors in memory could be using the Space Saving algorithm (Metwally, Agrawal and El Abbadi, 2005) together with the adaptive window (ADWIN) (Bifet and Gavaldà, 2007) change detector as was done in (Bifet et al., 2011).

The lexicon induction would be conducted by training an incremental word classifier using the vectorial representation of the words to form the feature space and a seed lexicon $\mathcal{L}$ to label the training instances. A strong assumption to be taken here is that the words from the seed lexicon do not change their polarity over time. The word sentiment classifier would be trained incrementally using stochastic gradient descent (SGD) as was done in Chapter 3, and would be used to classify the polarity of the unlabelled words observed in the stream. A pseudo-code of the process is given in Algorithm 7.3.1.

Algorithm 7.3.1: Algorithm for training an incremental polarity lexicon

```
input : tweetStream, L
foreach tweet ∈ tweetStream do
    words←tokenise(tweet)
    foreach word ∈ words do
        updateVector(word)
        if hasWord(word, L) then
        updateClassifier(getVector(word),getLabel(word, L))

    end
end
```

In order to track how the polarity of words changes over time we would need to periodically classify the unlabelled words. A simple approach is to classify (or re-classify) words every time they appear in a tweet. Alternatively, we could classify all the unlabelled words at regular intervals. A more sophisticated approach would be to use a change detector, such as ADWIN (Bifet and Gavaldà, 2007) for each word vector and reclassify words after detecting a change.

7.3.3 Extensions to ASA

ASA is a lexicon-based distant supervision model that enables the transfer of sentiment labels from the word-level to the message-level. Therefore, it could potentially be used for classifying tweets according to other sentiment labels associated with words, such as subjectivity labels, numerical scores indicating sentiment strength, and multi-label emotions.

Considering that ASA can generate large amounts of training data from large source corpora, it could also be suitable for training deep neural networks that learn more sophisticated representations of tweets for sentiment classification.

Another important aspect of ASA is its flexibility: it can be used with any kind of features for representing the tweets. For example, paragraph vector-embeddings (Le and Mikolov, 2014), which have shown to be powerful representations for sentences, could be trained from large corpora of unlabelled tweets and included in the feature space.

Finally, ASA could also be adapted for training incremental polarity classifiers in an on-line fashion from a stream of time-evolving tweets. An incremental version of ASA would dynamically add tweets with positive and negative words to the sets *posT* and *negT*. These sets would incorporate a forgetting mechanism to discard old tweets. The training instances would be generated by periodically sampling and averaging tweets from *posT* and *negT*, and these instances would then be fed to an incremental polarity classifier trained using SGD or other incremental learning approaches.

This approach could be used for online opinion mining from social media streams (Bifet and Frank, 2010), and would potentially be useful for tracking public opinion regarding high-impact events on Twitter, such as political campaigns, sports competitions, movie releases and natural disasters.

Bibliography

Akcora, C. G., Bayir, M. A., Demirbas, M., Ferhatosmanoglu, H. (2010). Identifying breakpoints in public opinion. In *Proceedings of the First Workshop on Social Media Analytics*, pp. 62–66. New York, NY, USA: ACM.

Amir, S., Ling, W., Astudillo, R., Martins, B., Silva, M. J., Trancoso, I. (2015). Inesc-id: A regression model for large scale twitter sentiment lexicon induction. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pp. 613–618. Association for Computational Linguistics.

Årup Nielsen, F. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the 1st Workshop on Making Sense of Microposts (#MSM2011)*, pp. 93–98.

Asur, S., Huberman, B. A. (2010). Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1, pp. 492–499. Washington, DC, USA: IEEE Computer Society.

Aue, A., Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. Tech. rep., Microsoft Research.

Baccianella, S., Esuli, A., Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pp. 2200–2204. European Language Resources Association.

Baeza-Yates, R. A., Rello, L. (2011). How bad do you spell?: The lexical quality of social media. In *Workshop on the Future of the Social Web*, pp. 2–5. Association for the Advancement of Artificial Intelligence.

Bahrainian, S. A., Liwicki, M., Dengel, A. (2014). Fuzzy subjective sentiment phrases: A context sensitive and self-maintaining sentiment lexicon. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences*

on Web Intelligence (WI) and Intelligent Agent Technologies, pp. 361–368. IEEE Computer Society.

Baroni, M., Dinu, G., Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 238–247. Association for Computational Linguistics.

Becker, L., Erhart, G., Skiba, D., Matula, V. (2013). Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises*, pp. 333–340.

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., Jurafsky, D. (2004). Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pp. 20–27. AAAI Press.

Bifet, A., Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th international conference on Discovery science*, pp. 1–15. Springer Berlin Heidelberg.

Bifet, A., Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the Seventh SIAM International Conference on Data Mining*, pp. 443–448.

Bifet, A., Holmes, G., Pfahringer, B. (2011). MOA-TweetReader: real-time analysis in twitter streaming data. In *International Conference on Discovery Science*, pp. 46–60. Springer Berlin Heidelberg.

Blei, D. M., Ng, A. Y., Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Bollen, J., Mao, H., Zeng, X.-J. (2011). Twitter mood predicts the stock market. *Journal of computational science, 2*(1), 1–8.

Bradley, M. M., Lang, P. J. (1999). Affective Norms for English Words (ANEW) Instruction Manual and Affective Ratings. Tech. rep., The Center for Research in Psychophysiology.

Bravo-Marquez, F., Mendoza, M., Poblete, B. (2013). Combining strengths, emotions and polarities for boosting twitter sentiment analysis. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, pp. 10–19. ACM.

Bravo-Marquez, F., Mendoza, M., Poblete, B. (2014). Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, *69*, 86 – 99.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, *18*(4), 467–479.

Calais Guerra, P. H., Veloso, A., Meira Jr, W., Almeida, V. (2011). From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 150–158. ACM.

Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems*, *31*(2), 102–107.

Cambria, E., Hussain, A. (2015). *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Cham, Switzerland: Springer International Publishing.

Cambria, E., Livingstone, A., Hussain, A. (2012). The hourglass of emotions. In *Cognitive Behavioural Systems*, vol. 7403 of *Lecture Notes in Computer Science*, pp. 144–157. Springer Berlin Heidelberg.

Carvalho, P., Sarmento, L., Silva, M. J., de Oliveira, E. (2009). Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pp. 53–56. New York, NY, USA: ACM.

Castellucci, G., Croce, D., Basili, R. (2015). Acquiring a large scale polarity lexicon through unsupervised distributional methods. In *International Conference on Applications of Natural Language to Information Systems*, pp. 73–86. Springer Berlin Heidelberg.

Ceron, A., Curini, L., Iacus, S. M., Porro, G. (2014). Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to italy and france. *New Media & Society*, *16*(2), 340–358.

Chang, C.-C., Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1–27.

Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357.

Church, K. W., Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, *16*(1), 22–29.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*, 37–46.

Collobert, R., Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM.

Crammer, K., Singer, Y. (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265–292.

Das, D., Kolya, A. K., Ekbal, A., Bandyopadhyay, S. (2011). Temporal analysis of sentiment events: a visual realization and tracking. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing*, pp. 417–428. Springer Berlin Heidelberg.

De Choudhury, M., Sundaram, H., John, A., Seligmann, D. D. (2008). Can blog communication dynamics be correlated with stock market activity? In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pp. 55–60. ACM.

Dodds, P., Danforth, C. (2010). Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. *Journal of Happiness Studies*, *11*(4), 441–456.

Durant, K. T., Smith, M. D. (2007). The impact of time on the accuracy of sentiment classifiers created from a web log corpus. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pp. 1340–1346. AAAI Press.

Ekman, P. (1992). An argument for basic emotions. *Cognition & emotion*, *6*(3-4), 169–200.

Engström, C. (2004). *Topic Dependence in Sentiment Classification*. Master's thesis, University of Cambridge.

Esuli, A., Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pp. 617–624. ACM.

Esuli, A., Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation*, pp. 417–422. European Language Resources Association.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Foster, J., Cetinoglu, O., Wagner, J., Le Roux, J., Nivre, J., Hogan, D., van Genabith, J. (2011). From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 893–901. Asian Federation of Natural Language Processing.

Gayo-Avello, D. (2011). Don't turn social media into another 'literary digest' poll. *Communications of the ACM*, *54*(10), 121–128.

Gayo-Avello, D. (2013). A meta-analysis of state-of-the-art electoral prediction from twitter data. *Social Science Computer Review*, *31*(6), 649–679.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 42–47. Association for Computational Linguistics.

Glorot, X., Bordes, A., Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 513–520.

Go, A., Bhayani, R., Huang, L. (2009). Twitter sentiment classification using distant supervision. Tech. rep., Stanford University.

Gonçalves, P., Araújo, M., Benevenuto, F., Cha, M. (2013). Comparing and combining sentiment analysis methods. In *Proceedings of the First ACM Conference on Online Social Networks*, pp. 27–38. ACM.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pp. 424–438.

Guerra, P. C., Meira, W., Jr., Cardie, C. (2014). Sentiment analysis on evolving social streams: How self-report imbalances can help. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 443–452. ACM.

Guo, W., Li, H., Ji, H., Diab, M. T. (2013). Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 239–249. Association for Computational Linguistics.

Hamilton, W. L., Clark, K., Leskovec, J., Jurafsky, D. (2016). Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 595–605. The Association for Computational Linguistics.

Harris, Z. (1954). Distributional structure. *Word, 10*(23), 146–162.

Hatzivassiloglou, V., McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 174–181. Association for Computational Linguistics.

Hatzivassiloglou, V., Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, pp. 299–305. Stroudsburg, PA, USA: Association for Computational Linguistics.

Hu, M., Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177. New York, NY, USA: ACM.

Hu, Y., Talamadupula, K., Kambhampati, S. (2013). Dude, srsly?: The surprisingly formal nature of twitter's language. In *Proceedings of the Seventh International Conference on Weblogs and Social Media*, pp. 244–253. AAAI Press.

Jansen, B. J., Zhang, M., Sobel, K., Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, *60*(11), 2169–2188.

Japkowicz, N., Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429–449.

Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 151–160. Association for Computational Linguistics.

Jungherr, A., Jurgens, P., Schoen, H. (2011). Why the Pirate Party Won the German Election of 2009 or The Trouble With Predictions: A Response to Tumasjan, A., Sprenger, T. O., Sander, P. G., & Welpe, I. M. "Predicting Elections With Twitter: What 140 Characters Reveal About Political Sentiment". *Social Science Computer Review*, pp. 1–6.

Jurafsky, D., Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice Hall, 2nd edn.

Kamps, J., Marx, M., Mokken, R. J., De Rijke, M. (2004). Using WordNet to Measure Semantic Orientation of Adjectives. In *Proceedings of the International Conference on Language Resources and Evaluation*, vol. 4, pp. 1115–1118. European Language Resources Association.

Kim, S.-M., Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Kiritchenko, S., Zhu, X., Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, *50*, 723–762.

Koppel, M., Schler, J. (2006). The importance of neutral examples for learning sentiment. *Computational Intelligence*, *22*(2), 100–109.

Kouloumpis, E., Wilson, T., Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg! In *Fifth International AAAI Conference on Weblogs and Social Media*, vol. 11, pp. 538–541. AAAI Press.

Ladha, K. K. (1993). Condorcet's jury theorem in light of de Finetti's theorem. *Social Choice and Welfare*, *10*(1), 69–85.

Le, Q. V., Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, pp. 1188–1196.

Li, S., Wang, Z., Zhou, G., Lee, S. Y. M. (2011). Semi-supervised learning for imbalanced sentiment classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 1826–1831. AAAI Press.

Li, T., Zhang, Y., Sindhwani, V. (2009). A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 244–252. Stroudsburg, PA, USA: Association for Computational Linguistics.

Lin, C., He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 375–384. New York, NY, USA: ACM.

Lin, C.-J., Weng, R. C., Keerthi, S. S. (2008). Trust region newton method for logistic regression. *Journal of Machine Learning Research*, *9*, 627–650.

Liu, B. (2009). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer.

Liu, B. (2010). Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*. Boca Raton, FL: CRC Press, Taylor and Francis Group.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Liu, K.-L., Li, W.-J., Guo, M. (2012). Emoticon smoothed language models for twitter sentiment analysis. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 1678–1684. AAAI Press.

Liu, Y., Huang, X., An, A., Yu, X. (2007). ARSA: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 607–614. ACM.

Logunov, A., Panchenko, V. (2011). Characteristics and predictability of twitter sentiment series. In *19th International Congress on Modeling and Simulation–Sustaining Our Future: Understanding and Living with Uncertainty*.

Manning, C. D., Raghavan, P., Schütze, H. (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pp. 171–180. ACM.

Melville, P., Gryc, W., Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1275–1284. ACM.

Metaxas, P. T., Mustafaraj, E., Gayo-avello, D. (2011). How (not) to predict elections. In *Proceedings of Third International Conference on Social Computing*, pp. 165–171. IEEE Computer Society.

Metwally, A., Agrawal, D., El Abbadi, A. (2005). Efficient computation of frequent and top-k elements in data streams. In *International Conference on Database Theory*, pp. 398–412. Springer Berlin Heidelberg.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119. Curran Associates, Inc.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, *3*, 235–244.

Mintz, M., Bills, S., Snow, R., Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint*

Conference on Natural Language Processing of the AFNLP, pp. 1003–1011. Stroudsburg, PA, USA: Association for Computational Linguistics.

Mishne, G., de Rijke, M. (2006). Moodviews: Tools for blog mood analysis. In *AAAI Symposium on Computational Approaches to Analysing Weblogs*, pp. 153–154. AAAI Press.

Mishne, G., Glance, N. (2006). Predicting movie sales from blogger sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs*, pp. 155–158. AAAI Press.

Mishne, G., de Rijke, M. (2006). Capturing global mood levels using blog posts. In *AAAI Symposium on Computational Approaches to Analysing Weblogs*, pp. 145–152. AAAI Press.

Mohammad, S., Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, *29*(3), 436–465.

Mohammad, S. M. (2012). #Emotional tweets. In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, pp. 246–255. Association for Computational Linguistics.

Mohammad, S. M., Kiritchenko, S. (2015). Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, *31*(2), 301–326.

Mohammad, S. M., Kiritchenko, S., Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises*, pp. 321–327. Association for Computational Linguistics.

Nadeau, C., Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, *52*(3), 239–281.

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., Wilson, T. (2013). Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises*, pp. 312–320. Atlanta, Georgia, USA: Association for Computational Linguistics.

Nigam, K., McCallum, A., Mitchell, T. (2006). Semi-supervised text classification using em. *Semi-Supervised Learning*, pp. 33–56.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., Smith, N. A. (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pp. 122–129. AAAI Press.

O'Reilly, T. (2007). What is web 2.0: Design patterns and business models for the next generation of software. *Communications & strategies*, pp. 17–37.

Pak, A., Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pp. 1320–1326. European Language Resources Association.

Pan, S. J., Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Pang, B., Lee, L. (2005). 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*, pp. 115–124. Stroudsburg, PA, USA: Association for Computational Linguistics.

Pang, B., Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, *2*, 1–135.

Pang, B., Lee, L., Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pp. 79–86. Association for Computational Linguistics.

Parrot, W. G. (2001). *Emotions in social psychology: Essential readings*. Psychology Press.

Pennington, J., Socher, R., Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543. Association for Computational Linguistics.

Petrović, S., Osborne, M., Lavrenko, V. (2010). The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pp. 25–26. Stroudsburg, PA, USA: Association for Computational Linguistics.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. Rep. MSR-TR-98-14, Microsoft Research.

Plutchik, R. (2001). The nature of emotions human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American Scientist*, *89*(4), 344–350.

Poria, S., Gelbukh, A. F., Cambria, E., Hussain, A., Huang, G. (2014). EmoSenticSpace: A novel framework for affective common-sense reasoning. *Knowledge-Based Systems*, *69*, 108–123.

Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pp. 43–48. Stroudsburg, PA, USA: Association for Computational Linguistics.

Read, J., Pfahringer, B., Holmes, G., Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, *85*(3), 333–359.

Salton, G., Wong, A., Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Schütze, H., Pedersen, J. (1993). A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*, pp. 104–113.

Severyn, A., Moschitti, A. (2015a). On the automatic learning of sentiment lexicons. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1397–1402. Association for Computational Linguistics.

Severyn, A., Moschitti, A. (2015b). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 959–962. New York, NY, USA: ACM.

Silva, I. S., Gomide, J., Veloso, A., Meira, W., Jr., Ferreira, R. (2011). Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 475–484. New York, NY, USA: ACM.

Silva, N. F. F. D., Coletta, L. F. S., Hruschka, E. R. (2016). A survey and comparative study of tweet sentiment analysis via semi-supervised learning. *ACM Computing Surveys*, *49*(1), 15:1–15:26.

Sindhwani, V., Melville, P. (2008). Document-word co-regularization for semi-supervised sentiment analysis. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 1025–1030. Washington, DC, USA: IEEE Computer Society.

Sintsova, V., Pu, P. (2016). Dystemo: Distant supervision method for multi-category emotion recognition in tweets. *ACM Transactions on Intelligent Systems and Technology*, *8*(1), 13:1–13:22.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics.

Speriosu, M., Sudan, N., Upadhyay, S., Baldridge, J. (2011). Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pp. 53–63. Stroudsburg, PA, USA: Association for Computational Linguistics.

Stone, P. J., Dunphy, D. C., Smith, M. S., Ogilvie, D. M. (1966). *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA: MIT Press.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, *37*(2), 267–307.

Tang, D., Wei, F., Qin, B., Liu, T., Zhou, M. (2014a). Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pp. 208–212. Dublin, Ireland: Association for Computational Linguistics.

Tang, D., Wei, F., Qin, B., Zhou, M., Liu, T. (2014b). Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *Proceedings of the 25th International Conference on Computational Linguisticss*, pp. 172–182. Association for Computational Linguistics.

Thelwall, M., Buckley, K., Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology, 63*(1), 163–173.

Tsai, A.-R., Wu, C.-E., Tsai, R.-H., Hsu, J. (2013). Building a concept-level sentiment dictionary based on commonsense knowledge. *Intelligent Systems, IEEE, 28*(2), 22–30.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., Welpe, I. M. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pp. 178–185. The AAAI Press.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417–424. Stroudsburg, PA, USA: Association for Computational Linguistics.

Turney, P. D., Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*(1), 141–188.

Valitutti, R. (2004). Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pp. 1083–1086. European Language Resources Association.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM.

Vo, D. T., Zhang, Y. (2016). Don't count, predict! an automatic approach to learning sentiment lexicons for short text. In *The 54th Annual Meeting of the Association for Computational Linguistics*, pp. 219–224. Association for Computational Linguistics.

Weichselbraun, A., Gindl, S., Scharl, A. (2014). Enriching semantic knowledge bases for opinion mining in big data applications. *Knowledge-Based Systems, 69*, 78–85.

Wiebe, J., Bruce, R. F., O'Hara, T. P. (1999). 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, pp. 246–253. Association for Computational Linguistics.

Wiebe, J., Riloff, E. (2005). Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *Proceeding of CICLing-05, International Conference on Intelligent Text Processing and Computational Linguistics.*, vol. 3406 of *Lecture Notes in Computer Science*, pp. 475–486. Mexico City, MX: Springer-Verlag.

Wilks, Y., Stevenson, M. (1998). The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Natural Language Engineering*, *4*(02), 135–143.

Wilson, T., Wiebe, J., Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354. Association for Computational Linguistics.

Witten, I. H., Frank, E., Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington, MA: Morgan Kaufmann, 3 edn.

Wu, C., Tsai, R. T. (2014). Using relation selection to improve value propagation in a conceptnet-based sentiment dictionary. *Knowledge-Based Systems*, *69*, 100–107.

Wu, F., Huang, Y. (2015). Collaborative multi-domain sentiment classification. In *Proceedings of the 2015 IEEE International Conference on Data Mining*, pp. 459–468. IEEE Computer Society.

Yu, H., Hatzivassiloglou, V. (2003). 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*, pp. 129–136. Stroudsburg, PA, USA: Association for Computational Linguistics.

Yu, S., Kak, S. (2012). A Survey of Prediction Using Social Media. *ArXiv e-prints*.

Zaragoza, J. H., Sucar, L. E., Morales, E. F., Bielza, C., Larrañaga, P. (2011). Bayesian chain classifiers for multidimensional classification. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 2192–2197. AAAI Press.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., Liu, B. (2011). Combining lexicon-based and learning-based methods for twitter sentiment analysis. Tech. rep., Hewlett-Packard Development Company, L.P.

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 919–926. New York, NY, USA: ACM.

Zhou, Z., Zhang, X., Sanderson, M. (2014). Sentiment analysis on twitter through topic-based lexicon expansion. In Wang, H., Sharaf, M. (Eds.), *Databases Theory and Applications*, vol. 8506 of *Lecture Notes in Computer Science*, pp. 98–109. Springer International Publishing.

Zimmermann, M., Ntoutsi, E., Spiliopoulou, M. (2014). Adaptive semi supervised opinion classifier with forgetting mechanism. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 805–812. New York, NY, USA: ACM.

Zirn, C., Niepert, M., Stuckenschmidt, H., Strube, M. (2011). Fine-grained sentiment analysis with structural features. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pp. 336–344. Asian Federation of Natural Language Processing.