UMA ABORDAGEM NÃO SUPERVISIONADA BASEADA EM AUTO-APRENDIZAGEM PARA A COMBINAÇÃO DE MÉTODOS DE ANÁLISE DE SENTIMENTOS

PHILIPE DE FREITAS MELO

UMA ABORDAGEM NÃO SUPERVISIONADA BASEADA EM AUTO-APRENDIZAGEM PARA A COMBINAÇÃO DE MÉTODOS DE ANÁLISE DE SENTIMENTOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Fabrício Benevenuto de Souza.
Coorientador: Marcos André Gonçalves.

Belo Horizonte

Julho de 2017

PHILIPE DE FREITAS MELO

AN UNSUPERVISED APPROACH BASED ON SELF-LEARNING FOR THE COMBINATION OF SENTIMENT ANALYSIS METHODS

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais – Departamento de Ciência da Computação. in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Fabrício Benevenuto de Souza.
Co-Advisor: Marcos André Gonçalves.

Belo Horizonte

July 2017

Melo, Philipe de Freitas

M528u An Unsupervised Approach Based on Self-Learning for the Combination of Sentiment Analysis Methods / Philipe de Freitas Melo. — Belo Horizonte, 2017
xxv, 68 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.
Orientador: Fabrício Benevenuto de Souza.

Co-orientador: Marcos André Gonçalves.

1. Computação — Teses. 2. Análise de sentimento. 3. Mineração de opinião. 4. Processamento de Linguagem Natural. 5. Bootstrap (Estatística) I. Orientador. II. Coorientador II. Título.

CDU 519.6*73 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

An unsupervised approach based on self-learning for the combination of sentiment analysis methods

PHILIPE DE FREITAS MELO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. FABRÍCIO BENEVENUTO DE SOUZA - Orientador
Departamento de Ciência da Computação - UFMG

PROF. MARCOS ANDRÉ GONÇALVES - Coorientador
Departamento de Ciência da Computação - UFMG

PROF. DANIEL HASAN DALIP
Departamento de Computação - CEFETMG

PROF. MARCO ANTÔNIO PINHEIRO DE CRISTO
Departamento de Ciência da Computação - UFAM

Belo Horizonte, 23 de junho de 2017.

To my lovely girlfriend, supportive parents, and awesome friends.

Acknowledgments

First of all, I would like to thank my parents to support me in every step of my life and encourage me to go after my dreams. Also thanks to Marco, Dudu and Carolina, my brothers and sister, who accompanied me and helped me grow as a person. Without you, this journey would not be possible at all.

I also thank to Clara, for all patience and effective help during many nights of work. Thanks for all attention and care I received even when I could not return in an equally way. Your company made each challenge much easier than it could be.

Thanks to all of my friends. Thanks Luiz, my dear friend with whom I shared most of good and bad moments in the university. Thanks to Lucas, Manu, Sâmara, Maria, Madeira for the moments of joy, without these my life would not be so fun.

To my advisors, Fabrício Benevenuto and Marcos Gonçalves, thank you for the academic counseling, for all knowledge I acquired during the process, for every incentive and for believing in my potential as researcher.

Finally, I would like to thank the Department of Computer Science and all its staff for attention and commitment to keep all program working.

To all who contributes in any way to my accomplishments throughout life, thank you very much.

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

(Alan Turing)

Resumo

A análise de sentimentos se tornou uma ferramenta muito importante para análise de dados de mídia social. Existem vários métodos desenvolvidos para este campo de pesquisa, vários deles trabalhando muito diferentes uns dos outros, cobrindo aspectos distintos do problema e estratégias diversas. Apesar do grande número de técnicas existentes, não há uma única que se encaixe bem para todos os casos e diversas origens dos dados. Além disso, no caso de abordagens supervisionadas, pode ser muito difícil obter dados rotulados para estratégias que exigem treinamento, principalmente para novas aplicações. Neste trabalho, propomos combinar vários métodos populares de análise de sentimento do atual estado-da-arte e eficazes, por meio de uma estratégia não-supervisionada com uso de bootstrapping para classificação de polaridade. Nossa solução foi completamente testada considerando treze diferente conjuntos de dados em vários domínios, como opiniões de produtos, comentários e mídias sociais. Os resultados experimentais demonstram que o nosso método combinado (conhecido como 10SENT) melhora a eficácia da tarefa de classificação, mas mais importante, ele resolve um problema-chave no campo. Nosso método aparece consistentemente entre os melhores métodos em vários tipos de bases de dados, o que significa que ele pode produzir os melhores resultados (ou perto de melhor) em quase todos os contextos considerados, sem quaisquer custos adicionais. A nossa abordagem de auto-aprendizagem é também muito independente dos métodos base, o que significa que é altamente extensível incorporar qualquer novo método adicional que possa ser desenvolvido no futuro. Finalmente, investigamos duas abordagens de "transfer learning" e "active learning" para a análise de sentimento e mostramos o potencial dessas técnicas para melhorar nossos resultados.

Palavras-chave: Análise de Sentimento, Mineração de Opinião, Classificação Combinada, Aprendizado Não-supervisionado, Bootstrapping, Processamento de Linguagem Natural..

Abstract

Sentiment analysis has become a very important tool for analysis of social media data. There are several methods developed for this research field, many of them working very differently from each other, covering distinct aspects of the problem and distinct strategies. Despite the large number of existent techniques, there is no single one which fits well in all cases and data sources. Moreover, in case of supervised approaches, it may be very hard to get labeled data for strategies that demand training, mainly for a new application. In this dissertation, we propose to combine several very popular and effective state-of-the-practice sentiment analysis methods, by means of an unsupervised bootstrapped strategy for classification of polarity. Our solution was thoroughly tested considering thirteen different datasets in several domains such as opinions, comments, and social media. The experimental results demonstrate that our combined method (aka, 10SENT) improves the effectiveness of the classification task, but more important, it solves a key problem in the field. It is consistently among the best methods in many data types, meaning that it can produce the best (or close to best) results in almost all considered contexts, without any additional costs. Our self-learning approach is also very independent of the base methods, which means that it is highly extensible to incorporate any new additional method that can be envisioned in the future. Finally, we investigate a transfer learning and active learning approach for sentiment analysis and show the potential of this technique to improve our results.

Palavras-chave: Sentiment analysis, opinion mining, combined classification, unsupervised learning, bootstrapping..

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Nowadays, machines can execute a long and complex list of tasks in almost all fields of knowledge. This includes from the production of entertainment and fun to the most advanced researches in Science. Many of these activities, computers can perform with property, not only fast calculations and large memory storage, but computers are also very good at playing some games, identifying objects, and understanding languages. They even surpassed the humans skills in some of them [Douglas, 1978]. However, something machines still have a lot to improve is the recognition and discernment of human sentiments and emotions. When we give a computer the ability to distinguish the reactions of people towards an entity or understanding about someone's feelings, we have a large range of beneficial applications and opportunities to explore that could help and improve our life in society.

Sentiment analysis, sometimes called *opinion mining*, is an area of the Natural Language Processing field (NLP) which aims to extract and analyze subjective information from people's emotions, opinions, sentiments, reactions and attitudes towards something else. This information can be expressed in many forms and it can be contained in many sources of data such as web written texts.

This type of task can add value and bring solutions to a wide variety of problems, especially in the field of web and social networks. Knowing "what people think" has always been an important piece of information for most of us in the decision-making process [Pang and Lee, 2008], so the resulting object of sentiment analysis may represent a valuable information when properly worked. It can tell us a little more about the textual data available on the net and a little more about the user who wrote the document under review.

Online social media systems are places where people talk about everything, sharing their take or their opinions about noteworthy events. Not surprisingly, sentiment

analysis has become an extremely popular tool in several analytic domains, but especially on social media data. The number of possible applications for sentiment analysis in this specific domain is growing fast. Many of them rely on monitoring what people think or talk about places, companies, brands, celebrities or politicians [Hu and Liu, 2004; Oliveira et al., 2013; Bollen et al., 2010].

Therefore, this field of knowledge is of great interest to a wide range of researchers, not only from Computing, like areas such as HCI (Human Computer Interaction) and Social Network, but can also be applied in Sociology, Marketing and Advertising, Psychology, Economics and Political Science [Hutto and Gilbert, 2014]. Sentiment analysis usages include from the detection of sentiment contained in textual data shared by users on microblogs [Mohammad et al., 2013] to applications predicting price changes in the stock markets and even comparisons and relations of people's sentiment with the weather forecast [Hannak et al., 2012].

The task of creating an application in sentiment analysis, however, presents some problems that need to be overcome. In Pang and Lee [2008], an influential work on this area, four major challenges are exposed which must be surpassed in order to improve the development of any application in search of opinions or reviews: (i) how to know if the user is looking for the subjectivity of the target material or not, (ii) determine which parts of the source material are relevant in order to extract subjectivity, (iii) how to accurately identify the sentiments, opinions or characteristics within a fragment of text or document (iv) and how to represent this information in a brief and reasonable way. This work will particularly address the two final problems, since we already have the documents that should be analyzed. We seek to answer the question of what feelings they manifest and, also, to understand better how different methods produce their results, in order to take advantage of the different information they provide for the creation of an ensemble method that exploits the potential of each one.

In this context, in here, we propose 10SENT, an unsupervised learning approach for sentence-level sentiment classification that tells if a given piece of text (i.e. a tweet) is positive, negative, or neutral. Accordingly, many methods have been developed to deal with these problems, exploring different strategies to classify the sentiment of Web-based messages. However, recent efforts have demonstrated that there is no single method that always achieves the best prediction performance in all scenarios [Gonçalves et al., 2013; Ribeiro et al., 2016]. In order to obtain better results than existing methods, our approach consists of combining their classification outputs in a smarter way.

More importantly, our proposed approach aims at solving a key problem in this field, related to reduce prediction performance variability across different datasets. Our

strategy relies on using a bootstrapped learning classifier that creates a training set based on a combination of predictions provided by existing unsupervised methods. The intuition of this strategy is that if the majority of the methods label an instance as positive, it is likely to be positive, and it could be used to train a classifier. This self-learning step provides to our method a level of adaptability to the context of the texts, reducing prediction performance variability, a key aspect of an unsupervised approaches, as we shall see.

1.1 Motivation

Due to the enormous interest and applicability, many sentiment analysis methods were proposed in the last few years, including SentiStrength [**?**], VADER [Hutto and Gilbert, 2014], OpinionLexicon [Hu and Liu, 2004], Umigon [Levallois, 2013], and SO-CAL [Taboada et al., 2011]. In common, these methods are unsupervised tools and have been applied to identify sentiment polarity (i.e. positive, negative, and neutral) of short pieces of text like tweets, in which the subject discussed in the text is known *a prior*.

These tools, as well as many others that are all currently acceptable by the research community as the state-of-the-art is not well established yet. Recent efforts [Ribeiro et al., 2016] have shown that the prediction performance of these methods varies considerably from one dataset to another. For instance, in that study, Umigon was ranked in the first position in five datasets containing tweets and was among the worst in a dataset of news comments. Even among similar datasets, existing methods showed high variability in terms of their ranked positions.

This suggests that existing unsupervised approaches should be used very carefully, especially for unknown datasets. More importantly, it suggests that novel sentiment analysis methods should not only be superior to existing methods in terms of predictive performance, but its relative prediction performance should also vary minimally when used in different datasets and contexts.

Another important point here is that many methods propose a supervised approach. Although machine learning is a powerful tool for the classification of polarity, it could be costly or even impracticable for sentiment analysis tasks to get a previous labeled dataset for training a classifier due to data subjectivity. Thus, it is essential that a method can work in a scenario without any training available, in other words, the process should be possible in an unsupervised manner.

1.2 Objectives and Goals

The goal of this work is to enhance the accuracy of sentiment analysis through the ensemble of different methods merged into a single tool, which presents better performance than individual strategies. For doing this, we intend to know what are the best methods to use for the ensemble and if it is possible to combine them and obtain a better performance.

This research explores the sentiment analysis area by addressing existing methods in classification of document polarity, also developing and evaluating an advanced tool with the combination of the said established methods. The identification of these methods is a part of this work that aims at better understanding this field and each of its strategies adopted in order to combine them, highlighting its advantages during final results, so we could achieve a higher accuracy in data from different sources (e.g. reviews, microblogs, comments).

In this work, we aim to create an ensemble method with a coverage, the proportion of the dataset which the method presents a label output, bigger than the individual ones, but which will continue to have high accuracy. It is possible to observe that some of sentiment analysis methods, for example, have a high rate of accuracy, but low coverage in the dataset, while others have high rates of accuracy and coverage in one dataset, but are underperforming in another set. The strengths of each method, if combined in a cautious way, can lead to better results with good accuracy and coverage in a wide range of data.

Also, a large portion of lexicon dictionaries used by other methods are developed to specific applications and domains, or the lexicons are used for other purposes than sentiment analysis. We have as objective the development of a method that can perform well for sentiment analysis and also can be used for diverse kinds of contexts and sources of text messages without significantly loses in terms of accuracy and confidence. Other works emphasize the large variability of the results of the methods for different types of data, so it is a key question in our problem the development of stable methods across varied data sources.

Here, we present 10SENT as an unsupervised tool to detect polarity in messages in order to fulfill these tasks and gaps that still need to be solved for sentiment analysis. As a result, the main contribution of this work would be an easily deployable method that can produce results as good or better than the best single method for each dataset (which can vary a lot) in a completely unsupervised way, being much superior than other unsupervised solutions such as majority voting, and in some cases close to the best supervised ones.

Finally, as a second contribution, we intend to perform an investigation into other important features that can add to the final results and refine the algorithm. Aspects like transfer learning and self-learning were ascertained during the research to improve the result without extra labeling effort.

1.3 Our Approach

We organize the proposed work into three main separated but related tasks: (i) a vast study about current sentiment analysis methods to identifying, understanding and selecting the strategies that could be exploited; (ii) development of an ensemble unsupervised method of sentiment analysis that combines methods already established in literature and pointed in previous step; (iii) extension and adaptation of the proposed approaches to identify features and quality indicators that influence the performance of different methods of sentiment analysis.

In the very first steps of our work, we study and select a group of methods of sentiment analysis well established in literature. This task consisted of a series of tests and analysis of how each method performed when applied in different datasets. We used many datasets of different sizes and sources, which are all publicly available.

During the next step, we exploit a simple combination technique – *Majority Voting*. Preliminary combinations compared with individuals techniques gave us a general idea how far we can reach with this strategy. Later, we improved this baseline by adding weights to each methods. The weighted vote, however, is still a simple strategy of combination, but it brings more features to use during the next steps.

For the construction of the ensemble strategy, it is necessary to observe features from both methods and datasets that can be used to determine the kind and nature of the sentimental text we are dealing. Some datasets are derived from different domains and structures, for example in microblogs. While some texts could be full of emotive thoughts and sarcasms, others can be totally formal and objective. These differences can be determinant to build a stable method that not depend of the context of the messages that will be analyzed.

After this, we built a more complex strategy to combine all outputs from other methods to construct a consistent tool to classify polarity of sentiment analysis texts at sentence level. One way to achieve this objective is through self-learning, more specifically, a bootstrapping approach. It is a technique used to iteratively improve a classifier's performance. It uses the output of each method as a parameter to a classifier and adds high confidence items to improve its own classification model. In this

scenario, we discard the need of manually supervised training data, but still enhance the classification results with information added by other methods outputs.

Finally, 10SENT was tested by combining the top (best) ranked methods. 10SENT was evaluated with thirteen gold standard datasets containing social media data from different sources and contexts. Those datasets consist of different sets of labeled data annotated for positive, negative and neutral texts from social networks and from comments of news, videos, websites and blogs. Our approach showed to be statistically superior to (or at least ties with) the existing individual methods in most datasets. As a consequence, the approach obtained the best mean rank position considering all datasets. Thus, our experimental results demonstrate that this combined method not only improves significantly the effectiveness of the classification task for many datasets but its cross-dataset performance variability is minimal (maximum stability). In practical terms, this means that one can use our approach in any situation the base methods can be applied, without any extra cost (since it is unsupervised) and without the need to discover the best method for a given context, and still producing top-notch effectiveness in most situations.

1.4 Organization

The remainder of this dissertation is organized as detailed below:

Chapter 2: Background and Related Work. This chapter presents the main concepts related to the area of sentiment analysis, on which this work is supported. We also present a brief exposition of many methods proposed in the literature for the task of polarity detection of sentiment. In addition, it describes the main related works, that explore the context of combining sentiment analysis methods.

Chapter 3: Preliminary Approaches. In this chapter, we introduce the methodology developed for this work, including database used, selection of methods and majority voting and exhaustive weighted voting strategies. Finally, we describe important limitations and actions taken to attempt to minimize the related impact;

Chapter 4: 10SENT. We present in this chapter our combined unsupervised method developed to this project. We also describe the problem we aim to solve with this method and our approach during steps of development. Finally, our tests and results concerning the strategy adopted are presented.

Chapter 5: Active and Transfer Learning. In order to further improve the method, this chapter shows additional approaches used in 10SENT that can help in the unsupervised task for sentiment detection of polarity.

Chapter 6: Results. This chapter presents results obtained by conducting the quantitative data analysis, including a discussion and the comparative analysis of the findings. We also present some upperbounds approaches and compare to our method developed.

Chapter 7: Conclusions and Future Work. Finally, we discuss the main conclusions of this dissertation, highlighting the main contributions and perspectives Of future work.

Chapter 2

Background and Related work

In this chapter, some definitions and concepts relevant to the understanding of this work are shown and a brief description of some methods of sentiment analysis and other literature review is presented.

2.1 Definitions and Used Terminology

Sentiment Analysis, or Opinion Mining, became a wide spread field in recent years. With these advancements in science, some terminology emerged to explain or describe tools and objects used in this topic. Thus, here we highlight the concepts involved in the execution of this work and offered grounds to its accomplishment.

2.1.1 Sentiment

Sentiment, in terms of this work, is the underlying feeling attitude, assessment or emotion associated with an opinion. We can present this sentiment as a triple [Liu, 2015]: (y, o, i), where y is the type of sentiment, o is the orientation of the sentiment and i is the intensity of the sentiment.

Sentiment type: The kind of the sentiment could be classified in several different ways, as linguistic, psychological, and consumer research based. Here, we treat sentiment as the definition broadly used in consumer researches and sentiment analysis field, as described in [Chaudhuri, 2006]. This definition divided sentiment in two categories: *rational* and *emotional*.

While **rational sentiment** represents rational reasoning, tangible beliefs, and utilitarian attitudes, they express no emotions. For example, " *The voice of this phone*

is clear." implies a rational sentiment towards the phone.

Emotional sentiment are emotional and non-tangible responses. It is more related to psychological relationship between people and the entity. "*I love it all*" , "*I hate to wait in the line*", and "*I cried when my team lost the championship*" are examples that imply emotional sentiment.

Sentiment Orientation: The orientation of sentiment can be *positive*, *negative*, or *neutral*. By neutral, it means the object analyzed has no sentiment associated with. It can be further detailed in *polarity*.

Sentiment intensity: Sentiment can present different levels of intensity or strength. It means that some texts have the same sentiment, but they can differ in degrees of magnitude. Words like "*good*" and "*awesome*", "*bad*" and "*terrible*", or intensifiers like "*very*", "*extremely*", "*super*", for example, can be signs or clues of the intensity of a text. Although those words and signs of intensity can help during the process of sentiment discovering, here we choose to group these sentiments despite their levels and reduce the sentiment just to its orientation.

2.1.2 Polarity

Same as sentiment orientation, we define polarity as the level of positivity, negativity or neutrality in a text. There is no distinction between messages with same sentiment but with different polarities, so there are three different and unique labels an instance can receive: "*Positive*", "*Negative*" or "*Neutral*", even though some are more expressive than others.

The polarity indicates what kind of sentiment is being expressed and represents an attitude or emotion its author has towards the target [Liu, 2015]. They are related to states of humor, embedded in a particular message, such as surprise, anger or happiness. Thus, a feeling is determined from the identification of the polarity of a text [Silva et al., 2012].

2.1.3 Document Analysis

An opinion can be defined by identifying two elements in a document: the target of the opinion and the expressed sentiment about that target. In this case, a document is characterized by any fragment of natural language text [Tsytsarau and Palpanas, 2012].

The analysis of a text can be done in different levels of a document. When the size of the object in study decreases, more specific the classification will be, i.e., a sentence

classification is more specific than a classification of the whole text. Three levels of a text are widely used in literature for sentiment analysis area: document level, sentence level and aspect level.

Document-Level: In this level, there is a focus in the classification of the whole text or document. In other words, an entire document with a set of phrases and paragraphs with opinions or sentiments expressed by a single person is the object of analysis. For this context, an entity can be described as an object or a topic

Sentence-level: This level is more specific than document-level in classification. While the former studies all phrases of one document at same time, the sentence-level analyzes sentences individually as one document can have both positive and negative sentences. So, this kind of structure is used when it is more accurate to treat sentences individually.

Aspect-level: Different from the other two levels that usually have an unique sentiment for a entire entity, aspect-level is about identifying different aspects of one single entity in a text and attributing sentiment to each one. For example, in "*This car has an excellent engine, but its tires are old and worn out*", we can observe the sentence express an opinion about a "car", which is the entity, but it evaluates two different aspects of the "car": "engine" and "tires". While one aspect has a positive opinion, other has a poor evaluation, so instead of label the "car" with an unique sentiment, aspect-level is used when it seeks to identify the sentiment of each aspect of an entity.

In this work, we focus entirely on sentence-level. Since many sources of text available to us origin from social networks and comments of websites, the format of this data is very small and it has just few sentences for each object in analysis. Thus, it is preferred to work on sentence granularity. Moreover, another reason that lead us to sentence-level is that many other methods of sentiment analysis mentioned here also work in a sentence-level, so it is easier to compare and combine them.

2.1.4 Unsupervised Learning

Unsupervised machine learning is a task of classification of unlabeled instances using hidden structures from a source of data. Different of supervised approaches, an unsupervised method doesn't need a previous training data to infer a label during classification process.

Examples of labeled data given to an algorithm can help to find patterns or infer functions to describe other similar unlabeled data. It is very common to see methods that use a model of training-test algorithm in machine learning problems. But, some tasks of classification have few or even no labeled data at all for training the algorithm.

In these cases, it is necessary to use models that doesn't require the use of a training set. Some methods of clustering, as k-means, are well-known methods of unsupervised learning approach.

Although there is some labeled dataset for sentiment analysis task, the availability of this kind of data is very scarce, so we opted to chose an unsupervised model to classify polarity on this project.

During development of 10SENT, the method we present in this work, there is, in fact, a supervised step during the process, but, as the use of this method doesn't depend on a previous labeled dataset, we can say that it remains as an unsupervised tool. It happens because the training set used in this step is created by the algorithm itself with inferred labels from structures of dataset.

All other methods used are models already trained or other strategies that don't need a training set as well, so all parts used to develop this system are also unsupervised tools.

2.1.5 Learning and Lexical Methods

Current methods of sentiment analysis in web data are given in two main categories: the **lexical methods** and the **machine learning methods**. [Pang and Lee, 2008] provide an overview of an area defined as the basis used by various opinion-mining researchers.

The machine learning methods are the classifiers that use the supervised model with training and test with some database previously labeled with prediction classes [Pang et al., 2002].

In contrast, the lexical methods does not require prior training; instead, it measures based on a word list. These lists are lexical dictionaries, lists of terms associated with feelings or other specific characteristics, to measure the sentiment in a document.

Each category has its advantages for sentiment analysis, i. e., the machine learning tends to be more accurate, but the lexical approach has better generality [Zhou and Chaovalit, 2008; Turney and Littman, 2003].

While the learning methods present a labeled dataset and tunning parameter constraints that can be difficult to obtain, lexical ones rely solely on the list of words they use, only demanding data for testing and performance verification. However, this performance is totally linked with how well the method's dictionary can describe the data, that is, if it has a sufficient and comprehensive vocabulary for the analyzed data at the same time it is specific enough to capture the feelings of the context.

Although lexical methods do not required labeled data, it is difficult to create a unique dictionary that fits for different source of data and context. Most of the lexical dictionaries used are developed specifically for the application in which they were used, or are based on other lexical dictionaries already existing in the literature. Some of them are based on concepts of psychology, but with adjustments and modifications of vocabulary to suit the system proposal and data in which it is used.

2.2 Sentiment Analysis Methods

Several methods for sentiment analysis on the Web have been proposed in the literature, which makes the area quite widespread in academia. Along with the need to create new tools, the theme has spread and generated a lot of search content. Ribeiro et al. [2016] is one of the first works that put a big effort to list, group and compare those methods. Its results try to clarify the real performance of these methods for polarity prediction in a considerable amount of datasets. They also contribute to the area by providing an open API for accessing and comparing sentence-level sentiment analysis methods, widely used in this work.

[Araújo et al., 2014] and [Araújo et al., 2016] also presents a quite extensive work to group methods and provide them as an accessible tool for sentiment analysis. They introduce the iFeel[1] system, which is an open Web API that allows anyone on the Web to test the various sentiment analysis methods including 19 sentence-level techniques and a multilingual architecture in its 2.0 version.

Following, we present some well-known and consolidated methods in literature recently used for sentiment analysis. Some of them were used during combination and integration phase of this work.

2.2.1 SentiStrength

SentiStrength implements a combination of supervised learning techniques with a set of rules that impact the "force" of the feeling expressed by the algorithm. Based on machine learning, SentiStrength compares supervised and unsupervised classification methods [Thelwall, 2013]. For applications in the context of social networks, these already existing techniques are combined and new features are added to the expanded version of the dictionary LIWC [Tausczik and Pennebaker, 2010]. These features include both extra set of words and expressions as well as emoticons.

[1] `http://www.ifeel.dcc.ufmg.br`

2.2.2 VADER

The VADER method derives from *Valence Aware Dictionary for Sentiment Reasoning* and is developed to analyze the feeling of messages in the context of Twitter and others social medias without the need for previous training. [Hutto and Gilbert, 2014] It has a lexicon of feelings, which was built and generated from the AMT. It uses the *bag-of-words* model combined with several other techniques developed over the application that aim to better explore the format of texts, such as excessive punctuation, capital letters, conjunctions that invert the Polarity of feeling, among many other syntactic aspects.

2.2.3 Opinion Finder

The Opinion Finder from MPQA [Wilson et al., 2005] performs subjectivity analysis trough a framework with lexical analysis former and a machine learning approach latter. It is a system that processes documents and automatically identifies subjective sentences as well as various aspects of subjectivity within sentences, including agents who are sources of opinion, direct subjective expressions and speech events, and sentiment expressions. It has a preprocessing step with Parts Of Speech (POS) tagging and identification of features, then it uses classifiers for subjectivity and polarity of sentences.

2.2.4 Opinion Lexicon

OpinionLexicon [Hu and Liu, 2004], also known as Sentiment Lexicon, is a lexical-based method consisting of two lists with 2,006 positive words and 4,783 negative words. It includes slang, misspellings, morphological variants, and social-media markups. This method is focused in predicting polarity for product reviews.

2.2.5 EmoLex

The EmoLex or *NRC Word-Emotion Association Lexicon* is a lexical method with about 14,000 words in English and approximately 25,000 "senses". In this list, each word is associated with one of eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust). This method is based on a large dataset classified by AMT turkers and also words taken from General Inquirer. It is possible to classify a sentence as positive or negative from the division of the eight basic emotions

in two groups of positive and negative sentiments. The classification occurs by counting frequency and co-occurrence of the words in a message.

2.2.6 SentiWordNet

SentiWordNet [Esuli and Sebastiani, 2006] is a lexical resource based on the WordNet dictionary [Miller, 1995] and is a widely used tool in the context of opinion mining. The method works with groups of adjectives, nouns, verbs, etc, called WordNet synonym sets (synsets). The synsets are associated to three polarity scores (positive, negative and neutral), using a score obtained through the use of a machine learning method. It also includes a semi-supervised learning step and a random-walk step (in SentiWordNet 3.0[2]) for refining the scores.

2.2.7 LIWC

LIWC (Linguistic Inquiry and Word Count) [Tausczik and Pennebaker, 2010] is a tool used for text analysis that estimates emotional, cognitive and structural components of any text provided as input, based on the use of dictionaries containing words and categories associated with each one. The method compares the words from the input text with the ones of its own built-in dictionary and finds out the percentage of words that reflect different emotions, social concerns, thinking styles and even parts of speech. The tool is uniquely commercial [3] and provides optimized functions such as permission to include specific and/or custom dictionaries.

2.2.8 ANEW and AFINN

The Affective Norms for English Words (ANEW) [Bradley and Lang, 1999] consists of a very popular data set that contains normative emotional classification for several words in English, where each one is associated with one of the three feelings of the scale: pleasure, excitement and dominance.

AFINN [Nielsen, 2011] builds a Twitter based sentiment Lexicon including Internet slang and obscene words. AFINN can be considered as an expansion of ANEW, a dictionary created to provide emotional ratings for English words. ANEW dictionary rates words in terms of pleasure, arousal and dominance.

[2]The tool can be accessed at `http://sentiwordnet.isti.cnr.it/`.
[3]The software is available at `http://www.liwc.net/`.

2.2.9 Umigon

Umigon [Levallois, 2013] is a web application providing a sentiment analysis for tweets for 8 different languages. Its main goal is to detect if a tweet has a positive, neutral or negative tone by disambiguating sentences using lexicon with approximately a thousand words combined with heuristics to detect negations plus elongated words and hashtags.

2.2.10 Pattern.en

The pattern.en module [Smedt and Daelemans, 2012] is a Python toolkit that contains a part-of-speech tagger for English to identify different structures in a sentence as nouns, adjectives, and verbs. It is a programming package (toolkit) that deal with NLP, Web Mining and Sentiment Analysis. Sentiment analysis is provided through averaging scores from words in the sentence according to a bundle lexicon of adjectives (e.g., good, bad, amazing, irritating, ...), resulting in two output scores: for sentiment polarity (from positive to negative) and subjectivity (from objective to subjective).

2.2.11 SO-CAL

The Semantic Orientation CALculator (SO-CAL) [Taboada et al., 2011], uses a lexicon dictionary annotated with their semantic orientation and incorporates intensification and negation. Semantic orientation is used as the measure of subjectivity and opinion in a sentence.

This method creates a new Lexicon with multi-grams hand ranked with scale +5 (strongly positive) to -5 (strongly negative). It also implements two features: weighting and multiple cut-off. First is an option to assign different weights to sentences or portions of a text second allows for multiple cut-offs, it means the task of classification is easily extendable to n classes, instead of just two (positive and negative). The method also used Amazon's Mechanical Turk as validation for annotated data and included other text characteristics during the process, like part of speech tagging, negation and intensifiers.

2.2.12 Sentiment140 and Sentiment140 Lexicon

Sentiment140 [Go et al., 2009] (previously known as "Twitter Sentiment") is a paid tool for sentiment analysis that uses classifiers built from machine learning algorithms. It was proposed as an ensemble of three classifiers (Naive Bayes, Maximum Entropy,

and SVM) built with a huge amount of data containing emoticons collected directly from Twitter API [4] to classify instances as positive or negative.

Sentiment140 Lexicon is another different method. It is a lexicon dictionary based on the same dataset used to train the Sentiment140 [Mohammad et al., 2013]. This method consists in a dictionary of words associated with positive and negative sentiments, which contains 66,000 unigrams (single words), 677,000 bigrams (two-word sequence) and 480,000 of unigram–unigram pairs, unigram–bigram pairs, bigram–unigram pairs, or a bigram–bigram pairs.

2.2.13 Emoticons

This is one of the simplest approach methods for detecting polarity of a text between positive and negative. It sorts the messages based on the emoticons they contain. For this, a predetermined set of positive and negative emoticons is used with some of its popular variation [Gonçalves et al., 2013]. Since the popularization of online chats, emoticons have become a very popular way of expressing certain emotions in messages and have become a well-publicized tool in web media. Its popularity was such that some of these "symbols" were included in the English Oxford Dictionary. Although they are intimately connected with the sentiment expressed in the text, their coverage is very low due to low presence in texts, but the accuracy which it matches their classification is quite high. Despite the strong connection between Emoticons and the feelings of those who write them, which makes this method one of the most efficient, there is a low coverage of messages in general.

2.2.14 Stanford Recursive Deep Model

The Stanford Recursive Deep Model (SRDM) is a method for feelings mining proposed by [Socher et al., 2013]. The SRDM uses a base created from movie reviews with AMT labeling on a scale ranging from very negative to very positive, including neutral. The method uses a different approach in the classification called Recursive Neural Tensos Network (RNTN), which process all sentences and computes the interactions between them. One of its good points is that the method is able to evaluate words which invert the polarity of sentiment in a text, what is difficult to achieve with methods based only on the frequency and co-occurrence of words in a message.

[4] More information about the Twitter API can be found at `http://apiwiki.twitter.com/`.

2.2.15 SASA

Also based on machine learning, SASA (SailAil Sentiment Analyzer) [Wang et al., 2012] is an open source tool originally proposed as a method for analyzing 17,000 labeled tweets associated with the North American elections of 2012. The open source tool was evaluated in the Amazon Mechanical Turk (AMT) [Ama, 2005], where the messages were labeled as positive, negative, neutral, or indefinite, by *turkers*. It is based on the statistical model obtained from the classifier Naive Bayes on unigram features. It also explores emoticons and exclamations.

2.2.16 SenticNet

The SenticNet [Cambria et al., 2010] is a method that explores artificial intelligence and semantic web techniques for opinion mining and sentiment analysis. The method uses the bag-of-concepts model, instead of counting words co-occurrence frequency by applying a dimensionality reduction to infer the polarity of common sense concepts.

The SenticNet invests in a natural language processing techniques to create semantic meanings, in order to infer the polarity of texts in positive or negative at semantic level, not the syntactic only. It allows the sentiments to have dynamics concepts which change depending on the relations between clauses.

2.2.17 Happiness Index

The Happiness Index proposed in [Dodds and Danforth, 2010] is a sentiment scale that uses ANEW as the basis for calculating the punctuation for a text provided as input, indicating the "amount" of happiness that exists in that text in a scale between 1 and 9. It is composed by a collection of 1,034 words commonly used associated with their affective dimensions of valence, arousal, and dominance.

2.2.18 PANAS-t

The PANAS-t (Positive and Negative Affect Schedule) [Gonçalves et al., 2013] method consists of adapting the application of a psychometric scale of measuring feelings to the context of social networks (e.g. Twitter). It is based on an expanded version of the *Positive Affect Negative Affect Scale* (PANAS) scale developed in 1998 by Watson et al., where adjectives are associated with one from eleven moods such as surprise, fear or guilt, through a questionnaire answered by users.

2.3 Overview of Combination Approaches for Sentiment analysis

The variety of issues in the field can provide different kinds of strategies of combinations in order to solve obstacles. In this scenario, there are challenges, such as the lack of annotated data, that we address in our work. In a real application of sentiment analysis, it can be very hard to get previous labeled data to train a classifier. Accordingly, we propose an unsupervised solution to deal with this problem that combines other known methods, taking advantage of each approach.

The strategy of combination is widely known in machine learning community and it is well explored in the literature [Kotsiantis, 2007; Dietterich, 2000]. This strategy covers algorithms that, from a set of classifiers, it labels data based on their individuals learning forecasts. Overall, this technique is an interesting research opportunity because it allows one to fill the gaps and individual shortcomings of existing methods, but on the other hand can greatly increase the computational cost of the results obtained.

However, the idea of combining different sentiment analysis strategies has been only recently explored. Some methods considered here consist of combined lexicons, such as AFINN, but most of the existing literature on the combination of sentiment analysis involves a learning component. Next, we briefly review some of them.

Dang et al. [2010] took a first step on this direction by combining machine learning and semantic-orientation for product reviews classification. Unlike the machine learning approach, a semantically-oriented method does not require prior training, since it only needs to consider words expressing positive or negative sentiments. The process of combination consisted of the development of a lexicon-enhanced method to generate a set of positive and negative word measurements to use as new features. They extracted features of content and sentiment from both machine learning and semantic-oriented approaches. In sequence, a SVM classifier is trained by combining these features to predict negative and positive polarity in four different datasets.

El-Halees et al. [2011] uses a combined classification approach to improve opinion extraction for documents of the specific Arabic language domain. Unlike ours, that work is focused on document-level sentiment classification and combines three distinct components. The first is a lexicon-based opinion classifier used with a dictionary of positive and negative words from SentiStrength, one of the methods used as part of our combined method. Then, they apply maximum entropy algorithm using as a training set for the documents classified on the previous step, aiming to classify as many documents as possible. Finally, they attempt to find the k nearest neighbors

among training documents classified in the previous two phases. Different from our proposed technique, this work combine strategies in a pipeline methodology and does not use existing methods.

In a similar way to the previous work Prabowo and Thelwall [2009] proposed a new hybrid classification method based on the combination of different strategies. This paper combines a rule-based classification, supervised learning and machine learning into a new hybrid classification by applying classifiers in sequence. This method is tested on movie reviews, product reviews and MySpace comments. But as a supervised approach, a training set of labeled documents is necessary.

An effort by Zhang et al. [2011] explored an entity-level sentiment analysis method specific to the Twitter data. A sentiment analysis in the entity-level granularity provides sentiment associated with a specific entity in the data (e.g. about a single product). In that work, the authors combined lexicon and learning-based methods in order to increase the recall rate of individual methods. The method first adopts a lexicon based approach, this can give high precision, but low recall. To improve recall, then, additional data are identified automatically by exploiting the information in the result of the lexicon-based method. A classifier is then trained to assign polarities to the entities newly identified. Training data are the tweets labeled by of the lexiconbased method. Differently from our work, this method was proposed for the entity-level, while ours focus on a sentence-level granularity.

In other work, Mudinas et al. [2012] proposed *pSenti*, a method for Sentiment Polarity Classification and Sentiment Strength Detection developed as a combination of lexicon and learning approaches for documents at concept-level (semantic analysis of texts by means of web ontologies or semantic networks) trained with movie and software reviews datasets. The supervised machine learning component uses a Linear SVM, and it is not just responsible for the tasks of adjusting final sentiment values, but also to find more sentiment words and evaluate all features of the sentiment system, including semantic rules used to derive the final output.

Moraes et al. [2013] investigated approaches to detect the polarity of FourSquare tips using supervised (SVM, Maximum Entropy and Naïve Bayes) and unsupervised (SentiWordNet lexicon) learning. They also investigate hybrid approaches, developed as a combination of the learning and lexical algorithms. All techniques were tested separately and combined, but the authors did not obtain significant improvements in hybrid approach over the individual techniques for this particular domain.

Gonçalves et al. [2016] analyzed different datasets and considered supervised machine learning in the context of classifiers ensembles. Their methodology also consists of combining a set of different sentiment analysis method in a "off-the-self" strategy to

generate the ensemble method. Their results suggest that it is possible to obtain significant improvements with ensemble techniques depending on the domain and shows the resilience of this approach in terms of cross-domains datasets.

In a more recently effort on the ensemble direction, Gonçalves et al. [2013] evinces the power of the combination of some of the state-of-the-art methods. Its results show that even a simple approach like majority vote were more promising than the individual methods alone, but they do not deepen in a more complex strategy for combination.

2.4 Research Gap

It is clear that there are many strategies and solutions when we want to measure feelings and sentiments in a textual object, thousands of works were published in the last few years and new methods are created in that direction. Therefore, efforts to gather a large part of these methods and compare them [Ribeiro et al., 2016] are extremely important to enrich the sentiment analysis researches because of the huge amount of material available. Systems developed like iFeel, a public Web API for sentiment analysis [Araújo et al., 2014; Araújo et al., 2016], allow scientists easily use and compare well-known methods.

Since there are so many possible methodological approaches to dealing with this task of sentiment analysis, it sounds natural to use some of them together to combine their strengths and reach an enhanced result that overcomes some problems as coverage and instability. Many authors enhanced sentiment analysis field by combining classifiers, but differently from these, we propose a novel approach by combining a series of existing methods in a totally unsupervised and elaborated manner. Most of those works combine using non-specific classifiers as SVM or Naive Bayes that need a training set, while our base methods are unsupervised tools specifically developed to sentiment analysis task, which allow us to create a new method that doesn't require the labeled training to generate a model but still have good results and better than simple approaches like majority vote. Another major difference of our effort is that we evaluate using multiple labeled datasets, covering multiple domains and social media sources. This is critical for an unsupervised approach given that the performance of similar methods vary significantly from one dataset to another. As we shall see, our solution produced the most consistent results across all datasets and contexts.

Chapter 3

Preliminary Approaches

In this chapter, we discuss some preliminary approaches studied during the research. Here, it is presented all datasets used for the polarity detection of sentiment in text, as the popular methods used in literature that cover this kind of task. An investigation of the combination of these methods is also initialized during this experimental process.

Before we develop a new method, the methodology of this work explore basic baselines as initial parameter to deeper understanding of different techniques on sentiment analysis. The state-of-the-art methods widely used for detect polarity is our baseline to development of the field, also a majority voting strategy is a classic combination that need to be compared and explored for a new ensemble method. Next, we go further in each of those aspects understanding how they work.

3.1 Datasets

In our evaluation, we use 13 datasets of messages labeled as positive, negative and neutral from many domains, including messages from social networks, movie and product reviews, opinions and comments in news articles. These datasets were kindly shared by authors from reference Ribeiro et al. [2016]. Next, we provide a short description of each dataset considered.

Sentistrength_bbc: Contains comments on BBC news. This dataset with 1000 messages (99 positive, 653 negative and 248 neutral), was labeled by 3 non expert annotators, with 87% of agreement between them. The average number of phrases and words is 3,98 and 64,39 respectively.

Sentistrength_digg: This dataset is composed by messages from Digg, a news aggregator website that brings together links to news, podcast and videos uploaded by

the users and evaluated by them. It contains 1077 messages (210 positive, 572 negative and 295 neutral), labeled by 3 non expert annotators with 88% of agreement. The average number of phrases and words are 2.50 and 33.97 respectively.

Vader_nyt: This dataset is composed by comments from the New York Times website content. Some comments are directly related to the news they were inserted to. With and agreement of 88% and labeled by 20 annotators using AMT, it has 2204 positive comments, 2742 negative and 244 neutral. Each comment has an average of 1.01 phrases and 17.76 words.

Sentistrength_twitter: This dataset has a corpus of 4242 tweets divided in three categories: 1340 positives, 949 negatives and 1953 neutrals labeled by non experts annotators.

Aisopos_ntua: This is also a social network dataset of tweets. It was labeled by experts and it has 139 positive tweets, 119 negative and 222 neutral, in a total of 500 instances. Each message has an average of 15.1 words.

Nikolaos_ted: This dataset contains 839 comments of TED conferences, in which 318 are positive, 409 negative and 112 neutral. Labeled by 6 annotators with no expertise, it has an 82% agreement rate and and average of 1 phrase per comment, with each comment containing an average of 16,95 words.

Sentistrength_youtube: Consisting of YouTube video comments, this dataset with 3407 messages labeled by non expert annotators has 1665 comments labeled as positive, 767 as negative and 975 as neutral. The level of agreement is 90% and the average number of phrases and words are 1.78 and 17.68 respectively.

Sentistrength_myspace: This dataset is composed by Myspace posts, a social network that was once the most popular in the world. With 1041 posts, 702 of them were labeled as positive, while 132 were considered negative and 207 neutral by the 3 non expert annotators. Each post has an average of 2.22 phrases and 21.12 words.

Debate: This dataset contains 3238 tweets in which 730 were labeled as positive and 1249 as negative. The remaining 1259 were classified as neutral. It was the unique dataset built with a combination of Amazon Mechanical Turk (AMT) Labeling with Expert Validation. To achieve accurate ratings, they selected 200 random tweets to be classified by experts and compared with AMT results. It was also the only dataset with 100% of agreement between the annotators. Each tweet has an average of 1.86 phrases and 14.86 words.

Tweet_semevaltest: Composed by 6087 tweets, of which 2223 classified as positive, 837 as negative, and 3027 as neutral, this dataset was provided as a list of Twitter ID's due to the social network policies related to data sharing. While crawling the respective tweets, a small part of them could not be accessed, as they were deleted.

This dataset was labeled by 5 annotators from AMT. Each comment has an average of 1.86 phrases and 20.05 words.

Sentistrength_rw: This dataset composed by 1046 messages from Runners World Forum contains 484 positive labeled messages, 221 negative and 341 neutral. The labeling was conducted by 3 non experts annotators. It has an average of 4.79 phrases and 66.12 words per entry.

English_dailabor: It is a set compounded by 3771 messages of tweets, labeled by 3 annotators from AMT in 739 positives, 488 negatives and 2536 neutrals.

Sanders: This dataset is composed of 3737 tweets. They were labeled in 580 positive, 654 negative and 2503 neutral messages by an expert annotator. This corpus has a average of 1.6 phrases and 66.12 words per tweet.

Table 3.1 summarizes the details of these datasets used during the research. There are information about name, amount of messages, amount of negative, positive and neutral instances and some information about average size of phrases and words of each one.

Dataset	**Messages**	**Positives**	**Negatives**	**Neutrals**	**Average # of phrases**	**Average # of words**
Sentistrength_bbc	1,000	99	653	248	3.98	64.39
Sentistrength_digg	1,077	210	572	295	2.50	33.97
Vader_nyt	5,190	2,204	2,742	244	1.01	17.76
Nikolaos_ted	839	318	409	112	1	16.95
Sentistrength_youtube	3,407	1,665	767	975	1.78	17.68
Sentistrength_myspace	1,041	702	132	207	2.22	21.12
Sentistrength_rw	1,046	484	221	341	4.79	66.12
debate	3,238	730	1249	1259	1.86	14.86
Sentistrength_twitter	4,242	1,340	949	1953	1.77	15.81
English_dailabor	3,771	739	488	2,536	1.54	14.32
aisopos_ntua	500	139	119	222	1.90	15.44
sanders	3737	580	654	2503	1.60	15.03
tweet_semevaltest	6,087	2,223	837	3027	1.86	20.05

Table 3.1. Labeled datasets details.

3.1.1 Gold Standard

The gold standard is the label assigned as positive, negative or neutral in each dataset. As sentiment polarity can be a subjective label, one must concern about how datasets are labeled to be the most accurate possible so that it corresponds to the reality of the data being worked on.

All datasets used during this research are results of several researches with efforts produced by experts or non-experts evaluators. Previous studies suggest that both kinds of annotation are valid, so we can assume that non-expert labeling may be as

effective as annotations produced by experts for affect recognition, a very related task Snow et al. [2008].

Thus, our effort to build a large and representative gold standard dataset consisted of obtaining labeled data from trustful previous efforts that cover a wide range of sources and kinds of data. We also attempt to assess the "quality" of our gold standard in terms of the accuracy of the labeling process.

3.2 Individual Methods Analysis

As we have many methods available, one of the first steps is to compare and analyze each method individually. There are many tools freely available to sentiment analysis, so we have to narrow this research by some parameters in order to choose some methods to our experiments.

With those methods selected, we perform some preliminary tests to check results and performance of each method individually as the basis to start the research.

3.2.1 Free Available Softwares

Although we have several good methods at hand to use, some of these are closed paid softwares and frameworks and we chose not to use them for the combination, restricting just to totally free tools. Some methods have remarkable results for some datasets, but were not used as LIWC (2007 and 2015), Semantria and SenticNet, for example. Although SentiStrength also has a paid version, it has a free of charge academic license. For SentiStrenght we used the Java version from May 2013 in a package with all features of the commercial version.

3.2.2 The "*Best 10 Methods*" Selection

We also have to limit the amount of methods we will use for our experiments. Although we have preselected ten unsupervised tools for this work, it is important to emphasize this approach can be easily applicable to any number of methods and for different sentiment analysis methods that were not picked here. To help in this task of the selection of the best techniques in current state-of-art for sentiment analysis we would use, we limited methods by its performance to use only "top 10" best results. In Ribeiro et al. [2016], a great effort was done to analyze and compare many and the major methods used in the community creating a benchmark study for sentiment analysis field. Based on its results, some methods stood out among the others in terms

of accuracy, therefore we decided to pick ten of the methods with best results that fitted in our needs.

Finally, we list all 10(ten) methods chosen during our analyses: VADER [Hutto and Gilbert, 2014], AFINN [Nielsen, 2011], OpinionLexicon [Hu and Liu, 2004], Umigon [Levallois, 2013], SO-CAL [Taboada et al., 2011], Pattern.en [Smedt and Daelemans, 2012], Sentiment140 Lexicon [Mohammad et al., 2013], EmoLex [Mohammad and Turney, 2013], Opinion Finder [Wilson et al., 2005], and SentiStrength [Thelwall, 2013].

3.2.3 Methods Results and Comparison

We perform a set of tests with those methods along all datasets to have preliminary results. But those methods have different formats and parameters, therefore, it was necessary to put some efforts by modifying some methods and getting it all together. After all methods adjusted, results are calculated for each method in each dataset. As follows, the detailed process is presented.

3.2.3.1 Output Adaptations

Each method has it own way to exhibit its results and its own format of data. So we have to note that the output of each method can present drastic variations depending on the goal it was developed for and the approach it employs. As mentioned, we can have a overview about how each method developed a different strategy to their results. Emolex, for example, provides the association of each word with eight sentiments. The word "unhappy" for example is related to anger, disgust, and sadness and it is not related to joy, surprise, etc.

In order to compare all these methods together, we have to adapt some of these outputs to normalize the data format. We discretized all methods outputs to fit the idea of polarity, converting each instance as positive, negative or neutral.

For example, OpinionFinder, generates polarity outputs (-1,0, or 1) for each sentiment clue found in a sentence so that a single sentence can have more than one clue. We considered the polarity of a single sentence as the sum of the polarities of all the clues. As many already had three class output format, there wasn't many barriers during this conversion process.

3.2.3.2 Individual results

For each dataset, all methods were tested to investigate individual characteristics and features that could be used during combination. We can see at table 3.2 that despite some good results, the performance of most of sentiment analysis methods are still low (under 75%). Column and row "AVG" and "StdDev" show average and standard deviation of methods and datasets. With these results, it is possible to notice that still exists a large gap of research that we can explore and improve in prediction of sentiment polarity of sentences, principally when we look at deviation of each method and dataset.

We can also observe with those tests that a single method can vary substantially across the datasets. This means that there is not a single method that fits well for all kinds of datasets. Since the source of data varies in format and origins (tweets, comments of blogs and videos), even the best method for one dataset can present poor results in others. The combination of low correlated classifiers is an essential characteristic for an ensemble method because we can add their advantages and overcome their drawbacks. It is considerably easy to reach good results by combining methods when each one can bring singularities to the final product.

	Vader	Afinn	Opinion Lexicon	Umigon	so-cal	pattern	sent140	Emolex	Opinion Finder	Senti Strength	**AVG**	**STD DEV**
aisopos	58.639	47.009	47.649	**74.856**	52.469	63.104	43.458	43.018	43.332	45.770	**51.930**	**10.548**
semevaltest	57.839	56.027	54.917	**61.977**	56.038	47.499	29.801	43.563	50.044	48.189	**50.589**	**9.173**
dailabor	57.728	57.659	59.511	**66.461**	60.788	51.092	30.087	48.754	56.016	57.353	**54.545**	**9.884**
sst_youtube	**56.387**	51.109	47.263	54.303	55.594	53.831	39.231	41.275	41.518	49.605	**49.012**	**6.398**
sst_twitter	56.327	52.415	52.023	**58.610**	56.572	53.428	39.080	46.247	48.249	45.088	**50.804**	**6.084**
sst_myspace	**55.027**	52.312	40.352	50.506	48.859	50.542	41.429	37.172	35.954	40.761	**45.291**	**6.868**
sanders	54.042	52.024	50.592	**56.515**	54.072	50.610	27.593	44.781	45.872	51.172	**48.727**	**8.244**
sst_digg	**51.883**	45.453	43.998	51.619	51.554	47.805	39.521	42.058	43.125	40.964	**45.798**	**4.658**
nikolaos_ted	**49.938**	42.430	42.654	40.951	48.336	46.373	38.644	40.828	41.508	35.407	**42.707**	**4.414**
sst_rw	**48.553**	47.781	46.929	42.817	48.440	44.119	38.909	35.612	39.197	35.612	**42.797**	**5.164**
debate	**45.181**	39.075	39.887	39.384	44.509	40.430	39.403	37.965	34.851	30.787	**39.247**	**4.331**
vader_nyt	43.736	33.304	36.116	24.810	**44.556**	38.484	43.094	34.368	29.302	18.432	**34.620**	**8.558**
sst_bbc	41.718	40.947	44.546	41.533	**45.813**	38.269	29.696	43.994	42.461	40.294	**40.927**	**4.513**
AVG	**52.077**	**47.503**	**46.649**	**51.103**	**51.431**	**48.122**	**36.919**	**41.511**	**42.418**	**41.495**		
StdDev	**5.765**	**7.085**	**6.477**	**13.250**	**4.997**	**6.921**	**5.535**	**4.245**	**6.913**	**10.031**		

Table 3.2. Table of Macro-F1 results to some methods of sentiment analysis

In Figure 3.1 we can see the variation of each method across different datasets. As we can observe, the difference between the worst and best results of one single method can be more than 40 percent, which provides evidence of our hypothesis that there is no method stable enough and well fitted to all datasets and contexts.

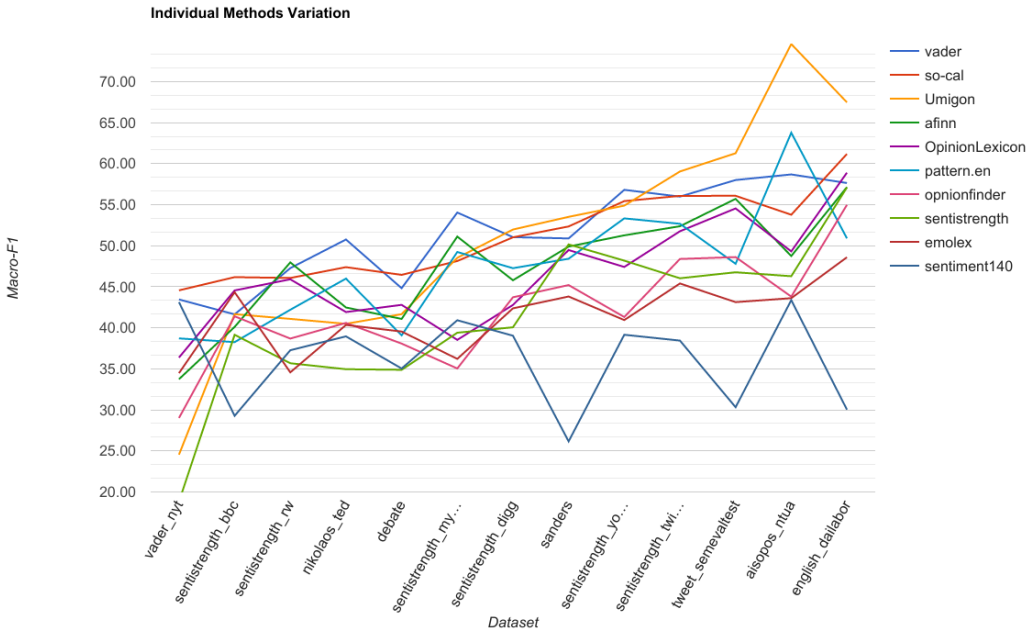

Figure 3.1. Chart showing variation of individual methods across all datasets

Besides the variation of methods across datasets, Figure 3.2 shows variation of methods to classify the different classes of the problem. This figure shows average F1 for each method by classifying positive, negative and neutral sentiment in a text. The first aspect important to note is that mean F1 for all classes is not very high, no more than 70% for all methods. It shows that all classes have difficulties that needs to be overcome during classification task of sentiment polarities. Other point is that methods can be very different from each other, as some are superior to label positive instances while others are better to classify neutral or negative sentiment but worse to other class. Thus, we can see that each method has its peculiarities that can sum to a single tool which perform a broad combination of them.

3.3 Majority Voting

Next, we will describe a baseline method we use to compare our proposed approach.

A Majority Voting strategy is a natural baseline. Voting is one of the simplest ways to combine several methods. By assuming that each individual method gives us a unique label as output for a sentence, the final result of majority voting is the label

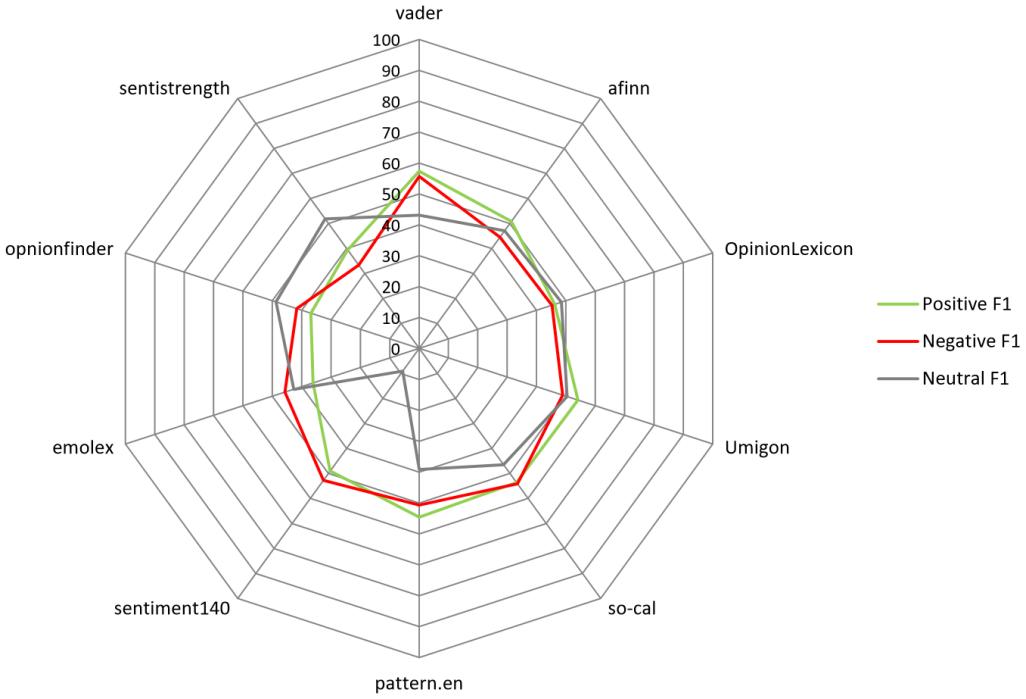

Figure 3.2. Average F1 measure by class for each method

which the majority of the base classifiers returned as output for that sentence. The major advantages of this approach are its simplicity and extensibility, i.e., it is very easy to include new (off-the-shelf) methods. Also, there is no need for training data for this method, which fits well in our purpose of an unsupervised method. On the other hand, it can not consider all the diversity of methods [Yeung et al., 2006].

More specifically, this combination works as follows: given an unlabeled instance x, the labels candidates set $L = \{w_1, w_2, w_3\}$ and the set of voting methods $M = \{m_1, m_2, m_3, ..., m_n\}$, we define a set V of votes v_{ij} of a sentence x as classes w_j given by the method m_i as follows:

$$v_{ij} = \begin{cases} 1, & \text{if method } i \text{ classified } x \text{ as class } j; \\ 0, & \text{otherwise.} \end{cases}$$

The final result R is given by the class j with maximum amount of votes:

$$R = argmax\left(\sum_{i=1}^{M} V_{ij}\right)$$

Because the format of this method, it is possible that two or more labels are the most voted class. To solve such cases, we assign a *Neutral* class to the cases of draw.

3.3.1 Analysis of Majority Results

For our approach, we chose 10 different methods during combination, but this strategy is not strict to this number, the quantity of methods integrated in combination can be higher or even smaller than 10 and it is easily adaptable to other variations. In order to see the impact of this choice on the results, the majority voting test was performed with different number of methods as showed in Figure 3.3.

	Number of Methods								
Dataset	2	3	4	5	6	7	8	9	10
english_dailabor	0.576	0.549	0.584	0.627	0.665	0.653	0.646	0.648	0.670
tweet_semevaltest	0.580	0.531	0.572	0.591	0.621	0.618	0.619	0.616	0.625
aisopos_ntua	0.587	0.480	0.504	0.540	0.587	0.607	0.643	0.619	0.593
sentistrength_twitter	0.560	0.500	0.536	0.548	0.579	0.581	0.595	0.593	0.591
sentistrength_youtube	0.568	0.500	0.530	0.530	0.560	0.555	0.561	0.548	0.547
sanders	0.509	0.471	0.512	0.512	0.545	0.530	0.540	0.540	0.542
sentistrength_myspace	0.540	0.492	0.507	0.499	0.519	0.524	0.536	0.517	0.519
sentistrength_digg	0.510	0.445	0.476	0.472	0.515	0.514	0.520	0.516	0.513
sentistrength_rw	0.473	0.432	0.470	0.470	0.484	0.479	0.489	0.477	0.485
nikolaos_ted	0.508	0.423	0.448	0.424	0.475	0.458	0.477	0.461	0.457
sentistrength_bbc	0.416	0.370	0.423	0.426	0.442	0.446	0.443	0.449	0.457
debate	0.448	0.382	0.427	0.408	0.437	0.437	0.453	0.447	0.442

Table 3.3. Test with different number of methods combined in Majority Voting

Besides the number of methods, some other features can be observed for this strategy, like number of correct votes, draws, agreement level and distribution of positive, negative and neutral votes.

Also, some methods produce an output of *"no answer"* for some instances. This is important to check the coverage of those methods for each dataset. In Table 3.4 we can see more detailed data for the majority results of ten combined methods.

The agreement level corresponds to how many methods agree with each other to produce the final decision. This information is important to check the confidence with which the majority voting classifies the dataset. As this number increases, the method can affirm more precisely that an instance is positive, negative or neutral.

The number of "*no answer votes*" shows the amount of instances that the most methods were not able to provide an effective output. This can be interpreted as the coverage of majority voting, because in those instances this method could not properly solve the polarity problem.

The coverage of majority voting is very high (more than 75% in all datasets) when compared with some individual methods, but for our approach we want to surpass

Dataset Name	%Correct Votes	%Neutral Votes	%Positive Votes	%Negative Votes	%Not Answer Votes	%Draws	%Average Agreement
english_dailabor	0.717	0.54	0.291	0.169	0.244	0.072	0.699
tweet_semevaltest	0.633	0.469	0.334	0.197	0.206	0.082	0.658
aisopos_ntua	0.57	0.409	0.347	0.244	0.201	0.09	0.667
sentistrength_twitter	0.587	0.482	0.316	0.201	0.218	0.074	0.682
sanders	0.614	0.548	0.273	0.179	0.291	0.072	0.699
sentistrength_youtube	0.541	0.407	0.383	0.21	0.19	0.082	0.696
sentistrength_digg	0.522	0.388	0.264	0.348	0.169	0.056	0.674
sentistrength_myspace	0.545	0.414	0.422	0.164	0.196	0.1	0.692
sentistrength_rw	0.502	0.282	0.412	0.307	0.118	0.067	0.636
nikolaos_ted	0.486	0.342	0.367	0.291	0.14	0.066	0.64
sentistrength_bbc	0.529	0.288	0.28	0.432	0.111	0.072	0.642
debate	0.436	0.483	0.255	0.262	0.213	0.081	0.653
vader_nyt	0.378	0.461	0.252	0.286	0.179	0.084	0.642

Table 3.4. An overview about votes by combining 10 methods with Majority Voting strategy

those numbers by always providing a polarity result for any given instance. To do it so, when a method has the *"NA"* label, it is converted to a Neutral polarity. We used this conversion based on idea that if it could not tell whether a sentence has a positive or negative sentiment it can be interpreted as neutral. This is an important heuristic defined in how the method works to guarantee that it will always have an output for a sentence, so that it can obtain a total coverage for all datasets.

3.4 Exhaustive Weighted Voting

We explore majority voting even further by adopting weights for each method. To develop a better combination we investigate the results of different influences of a method in the final decision during the combination.

This algorithm is a linear combination method. We exploit strategy that uses real labels of datasets to find the best possible combination of weights for each method we are working on. By doing this, we can know how far all these methods can perform together in a linear weighted strategy. In this sense, this an upperbound supervised baseline.

This method, like the majority voting, classifies an instance based on most votes each class receives. Distinct from the original Majority Voting method, in which all methods have the same weight, in here, one method can influence the final classification more than others.

The Exhaustive Weighted Voting works with weights found by means of an exhaustive search for each dataset. This search is performed exhaustively, in other words, it is performed evaluating every possible combination and seeking to maximize the clas-

sification result's in terms of Macro-F1 for each dataset.

During tests, we found that using five different weights from 0 to 1 was enough to estimate the best result for a weighted vote $W = \{0, 0.25, 0.5, 0, 75, 1\}$. While more than five weights did not imply in significantly better results, more weights mean a huge computational cost. Then, for each method, we associated a weight to its output and, as a result, the class with the highest weight was marked as resulting label of each instance.

. Table 3.5 presents the average weights and the standard deviation for each method in all datasets. We can see that most methods have a different behavior for each different data source. The same method may has a huge variance of weights in different datasets which can preclude the use of a unique weighted method. Despite this, we can observe that some methods have clearly a higher average than others even with a high deviation. This also indicates that just applying a simple majority voting may be not very effective.

	Weights				
	pattern.en	sentiment140	emolex	opinionfinder	sentistrength
Avg. Weight	0.28	0.37	0.26	0.40	0.85
Std. Deviation	0.25	0.28	0.29	0.31	0.28

	Weights				
	vader	afinn	OpinionLexicon	Umigon	so-cal
Avg. Weight	0.44	0.25	0.27	0.61	0.66
Std. Deviation	0.24	0.25	0.25	0.34	0.38

Table 3.5. Average and deviation for weights found during Exhaustive Weighted Vote step

Chapter 4

10SENT - A new combination approach for Sentiment Analysis

In here, we describe the developed combined approach for sentiment analysis, named 10SENT. First, we present the formal definition of the problem. Then, we describe the combined self-learning method. Finally, we present tests and discuss results, leading to all decisions made in this chapter.

4.1 Problem Definition and Scope

Sentiment analysis can be applied to different tasks. We restrict our focus on combining efforts related to detecting the polarity (i.e. positivity, negativity, neutrality) of a given short text (i.e. sentence-level). In other words, given a set S of opinionated sentences, we want to determine whether each sentence $s\ in\ S$ expresses a positive, negative or neutral opinion. We focus our effort on combining only unsupervised "off-the-shelf" methods and our strategy consists of using the output label predicted by each individual as input for a bootstrapping technique, a self-starting process supposed to proceed without external input.

4.2 Bootstrapping Approach

Our bootstrapping technique is an unsupervised machine learning algorithm which uses the sentiment scores produced by each individual sentiment analysis method to create a training set for a supervised machine learning algorithm. With this algo-

rithm, we are able to produce a final result regarding the sentiment of a sentence. Note that we did not need to use any labeled data in order to produce the model.

We describe this method in Algorithm 1. Suppose we have access to a set of sentences $S = \{s_{tr0}, s_{tr1}, s_{tr3}, ..., s_{tr_n}\}$ which are candidates of being part of our training data. Our goal is to use the unlabeled data S in order to produce a training set $train$ and, then, apply it to sentences which we want to predict (here represented as $test = \{s_{tst0}, s_{tst1}, s_{tst3}, ..., s_{tst_n}\}$), generating the predictions P. The training data $train$ is represented by a set of pairs (c, s) where c is the class representing a sentiment (positive, negative or neutral) obtained by using the information of each sentiment analysis method described in Section 3.2 and s is the sentence which is represented by a set of features which, in our case, is the sentiment method outputs. The $test$ is represented by a set of sentences $test = \{s_{tst0}, s_{tst1}, s_{tst3}, ..., s_{tst_n}\}$ and, the prediction P, contains a set of triplets $(s, predicted\_class, confidence)$ representing the sentence, the predicted class and the confidence (i.e. a score representing how confident the machine learning method is in its prediction), respectively.

We use the function $agree(s)$, for each sentence s, which computes the Agreement level, in other words, the maximum number of sentiment analysis methods agreeing each other regarding the sentiment in the sentence s. If this number is higher than the threshold A, we add the sentence s in the training set, removing it from S. Note that, when adding a sentence to $train$ we use the method $agreeClass(s)$ in order to obtain the class c which is the sentiment for s. Class c is obtained by using the class which has the majority of sentiment analysis methods assigned to the sentence s.

After doing this to all the sentences, in S remains instances which we could not discover a label to it, as the agreement was lower than the threshold. Then, in order to increase our training data, we use our training set $train$ to create a model and apply it in sentences S producing the predictions P. Thereby, we are able to use P adding more sentences to $train$. In order to avoid noise, we only add sentences where the supervised learning method give at least a certain confidence C that the prediction is right. Finally, with the training model $train$ created, we apply it to $test$ in order to produce, for each sentence, a single score c representing the sentiment score.

4.3 Off-the-shelf Methods

Our effort consists of combining popular "off-the-shelf" sentiment analysis methods freely available for use. It is important to highlight that the number of methods to be combined is not necessarily restricted to ten. In fact, there is no limit on the

Algorithm 1 Bootstrapping Algorithm

Require: Minimum of Agreement A
Require: Minimum of Confidence C
Require: The set of n sentences $S = \{s_0, ..., s_n\}$, candidates of being part of our training data
Require: The set of m sentences which we want to predict $test = \{s_0, ..., s_m\}$
Let $train$ = our training set represented by (c, s) which c is the target class and s is the sentence
Let P = our result which is represented by a set of triplet $(i, predicted\_class, confidence)$ which is the instance, the predicted class and its
for all $s \in S$ **do**
    if $agree(s) \geq A$ **then**
        Add the pair $(agreeClass(s), s)$ to $train$
        Remove s from S
Create a model M using $train$
Apply the model M in S to obtain the predictions P
for all $(s, predicted\_class, confidence) \in P$ **do**
    if $confidence \geq C$ **then**
        Add the pair $(predicted\_class, s)$ to $train$
Create a model M using $train$
Apply the model M in $test$ to obtain the predictions P

number of methods we can include as part of our approach – thus, we focus on the ones evaluated by Ribeiro et al. [2016]. We note that many sentence-level sentiment analysis methods combine a lexicon and a series of rule-based techniques to assess a sentence polarity. Many approaches make use of a series of intensifiers, punctuation transformation, emoticons, and many other heuristics. Some efforts explore supervised learning as part of their solution, but they make available an unsupervised tool. Some methods consist of simple lexical resources, not corresponding to sentence-level methods. Finally, the authors of that work also performed small adaptations in some methods to provide as output positive, negative and neutral decisions. We have used code shared by the authors of Ribeiro et al. [2016], who provide more details about implementations. The considered methods include: VADER [Hutto and Gilbert, 2014], AFINN [Nielsen, 2011], OpinionLexicon [Hu and Liu, 2004], Umigon [Levallois, 2013], SO-CAL [Taboada et al., 2011], Pattern.en [Smedt and Daelemans, 2012], Sentiment140 [Mohammad et al., 2013], EmoLex [Mohammad and Turney, 2013], Opinion Finder [Wilson et al., 2005], and SentiStrength [Thelwall, 2013].

We also note that all methods exploit light-weighted unsupervised approaches that rely on lexical dictionaries, usually implemented as a hash-like data structure. For this reason, the execution performance of our combined, as well as the individual methods, does not require any powerful hardware platform. There are even recent

efforts that ran some of these methods in smartphones. [Messias et al., 2016].

4.4 Evaluation Metrics

Next, we describe the metrics used to evaluate and compare the proposed methods.

4.4.1 Coverage

The first metric we consider is the coverage of a method in a dataset. Coverage is the proportion of instances in the dataset with which a method is able to detect some polarity. I.e., if a method gives the answer for a half of dataset, it has 50% of coverage. Even some methods with highly accurate results can present sometimes low coverage and don't produce an output sentiment score for a large quantity of sentences of the dataset, and it can interfere in overall results. So, it is important to assess the amount of instances that the method can cover. In table 4.1 we can observe coverage values of base methods for some datasets.

Although some base methods do not have full coverage, it is important to stress that both our proposed method – 10SENT – and the baselines have 100 percent coverage by definition, because they always give an output of positive, negative, or neutral independent of the sentence, therefore they present full coverage for all datasets as we can see at table.

Dataset	Vader	Afinn	Opinion Lexicon	Umigon	so-cal	pattern.en	sent140	Emolex	opnion finder	Senti-strength	10SENT
BBC	90.69	85.11	80.72	50.93	82.85	54.79	97.07	80.72	76.46	32.85	100.0
DIGG	82.23	74.81	69.44	63.04	71.99	56.65	89.64	67.14	56.27	27.49	100.0

Table 4.1. Coverage experiments results for 2 datasets

4.4.2 F1-Measure

A good method is expected to have a good effectiveness in its predictions. One of the most used effectiveness measures is the F1-Score. It can be calculated for each class by considering the following matrix for a single class:

		Actual observation	
		Positive	Negative
Predicted expectation	Positive	a	b
	Negative	c	d

Let a and d be the amount of correctly predicted instances as positive (true positive) and negative (true negative) respectively for an individual class, b denotes negative instances classified as positive (false positive), and c negative instances classified as positive (false positive). The precision rate is the proportion of retrieved instances that was correctly classified for this class, while recall is the proportion of real instances of this class that was retrieved from all dataset. Precision can be calculate as $p = a/(a + b)$ and recall as $r = a/(a + c)$. Then, the F-Score is the harmonic mean of precision p and recall r. It can be calculated for each single class as follows:

$$F_1 = 2 \cdot \frac{p \cdot r}{p + r}$$

We also use Macro-F1, or macro-average measure, to compute F1 score among all labels. This method can be used when it is wanted to know how the system performs overall across different classes. As we have three different classes (positive, negative and neutral), we calculate an average precision and recall between them as next equation:

$$(p_{macro}, r_{macro}) = \left(\frac{1}{q} \sum_{\lambda=1}^{q} p_\lambda, \frac{1}{q} \sum_{\lambda=1}^{q} r_\lambda \right)$$

Where λ is a label and $L = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ is the set of all labels. Finally, the Macro measure will be simply the harmonic mean as calculated in equation of F1-score using these two averages p and r.

4.4.3 Mean Ranking

As we have several outcomes, considering all base methods and datasets, it is important to have a global measure of performance for all these combinations in a single metric. For doing so, we ranked each method in each dataset and proposed a measure to assess the overall ranking performance. The Mean Ranking is the simple sum of ranks obtained by a method in each dataset divided by the total number of datasets, as below:

$$MeanRank(m) = \frac{\sum_{i=1}^{D} r_i}{|D|}$$

Where D is set of datasets and r_i is the rank of the method m for dataset i. It is important to notice that the rank was calculated based on Macro F1.

4.5 Experimental Setup

Our experiments were run using a 5-fold cross validation setup, with the best parameters for the learning methods found using cross-validation in the training set. This procedure was applied to all datasets enumerated in Section 3.1. We repeated this procedure 5 times, therefore, all the results reported next in this Chapter are the average of 25 test folds.

To compare the average results in the test sets of our experiments, we assess the statistical significance of our results by means of a paired t-test with 95% confidence, so we just consider statistically significant results which the value of p is less than 0.05 in our conclusions and any claim stated in this paper is based on results of these tests.

In the case of the Bootstrapping, there are two main parameters to set up: a level of agreement between methods in first phase, and a Confidence level in second phase to measure highly reliable classified items by learning algorithm. For each one, a series of tests were performed in the training sets to discover the best combination of them for our 10SENT proposal.

In the case of the base methods, the original output values (i.e. positive, negative or "zero" values) were considered as the corresponding polarities. In particular, an output equal to zero was considered as a neutral polarity or "absence of opinion". For outputs represented as a range of strengths or values (i.e. 1 to 5), we converted them to follow the same nominal pattern. Undefined answers by any method were considered as neutral, as done in Ribeiro et al. [2016].

We first analyze which machine learning technique is the best suited to our proposal since we rely on estimates of confidence. After that, we compare our results with unsupervised state-of-art base methods as well as the "stronger" exhaustive majority voting baseline. Finally, we discuss the possible upperbounds of an unsupervised method as ours and start some investigation on issues related to active and transfer learning for sentiment analysis and show the potential of these techniques to improve our results.

4.6 Choice of Classifier

10SENT is an unsupervised machine learning method as it does not exploit manually labeled data, only the agreement among the base methods, and given that the bootstrapping process adds a set of instances with high confidence into a training set, it is possible to perform a learning step using such data using the usual format train-

ing/validation. Because of this, there is a need to investigate which classifier fits better for this application.

Thus, we perform a series of tests with our method using different classification algorithms in order to choose the best algorithm for this task. In all these tests, we used all 10 methods of 10SENT.

We tested three different and widely used algorithms in our approach: *Support Vector Machine* (SVM) [Chang and Lin, 2011], *Random Forest* (RF) [Breiman, 2001] and *k-Nearest Neighbors* (KNN) [Pedregosa et al., 2011]. The SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification task by separating the training with maximum distance (margin) in different classes, in our case, positive, neutral and negative. The RF classifier is a variation of a bagging of decision trees. An ensemble of low-correlated decision trees built by a random attribute selection to compose the decision nodes. The KNN consists in a classification method which performs a selection of k closest training examples in the feature space to assign a label to a unlabeled instance.

Here we used the implementations of RF and KNN provided in scikit-learn[1] and for SVM, we use LibSVM[2] package, specifically, we are using *radial basis function* (RBF) kernel for experiments. Also, in order to optimize the parameter choice, we have used a grid search for the algorithm.

Table 4.2 shows some results (F1-Score) for a few datasets, but the results were similar for most of them. Overall, Random Forests produced the best results, being the final choice for our bootstrapping method.

DATASET	KNN	SVM	Random Forest
english_dailabor	55.5(± 1.7)	58.2(± 1.6)	60.9(± 1.7)
aisopos_ntua	51.7(± 4.3)	59.0(± 1.0)	59.7(± 2.5)
sentistrength_digg	46.0(± 1.6)	51.7(± 2.8)	49.4(± 1.0)
debate	40.6(± 1.5)	23.7(± 19.8)	42.6(± 1.7)
sentistrength_rw	37.3(± 5.0)	17.3(± 14.6)	34.6(± 1.8)

Table 4.2. Results of different classifier algorithms for learning step of 10SENT.

4.7 Choice of Number of Methods

Also, to verify the coherence with results obtained in Majority Voting method, we perform a test with different number of methods used in combination. In this test, we

[1]Available at `http://scikit-learn.org/stable/index.html`
[2]Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

want to check how an addition of a method can impact the outcome of the combination. Table 4.3 shows results of 10SENT combining from 3 up to 10 methods. In these experiments, we included from the best to the worst method in each dataset, according to Ribeiro et al. [2016]. As expected, adding a new method improves the overall results, but it is possible to note that improvements get lower with new inclusions. Thus, after a certain amount, the gain is minimal. Therefore, we fixed 10 as a good choice to number of methods in 10SENT core.

	#Methods							
DATASET	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
english_dailabor	49.90	65.68	68.26	66.38	67.09	68.68	66.70	69.68
aisopos_ntua	41.90	59.21	57.78	61.56	59.99	58.13	59.78	65.21
tweet_semevaltest	39.54	56.31	61.51	61.65	62.06	62.94	62.26	62.64
sentistrength_twitter	36.26	49.24	56.20	57.88	57.91	57.96	57.50	58.68
sentistrength_youtube	38.14	50.01	54.98	56.56	57.13	55.83	55.45	56.93
sentistrength_myspace	26.88	44.94	47.37	53.04	52.91	51.53	46.99	55.00
sanders	46.40	54.22	54.34	54.59	55.84	57.14	54.30	53.03
sentistrength_digg	28.16	43.62	48.67	50.44	50.59	52.47	51.55	54.18
sentistrength_rw	30.47	42.60	50.37	48.30	49.55	48.61	47.39	47.30
sentistrength_bbc	21.92	35.79	45.35	47.15	48.41	49.45	47.34	45.72
debate	25.10	34.03	41.40	45.76	45.17	45.11	42.74	45.06
nikolaos_ted	24.42	34.82	41.32	42.44	46.19	44.11	44.80	42.56
vader_nyt	9.38	19.64	30.27	34.97	37.42	36.83	36.98	37.97

Table 4.3. Test with 10SENT varying number of methods used in combination.

4.8 Choice of Parameters

In our method, we need to define two important parameters: the agreement level and the confidence level. Accordingly, we performed a study to understand better how our method performs when varying such parameters. In more details, the first tested parameter was the minimum number of agreements among the methods we should use in the first round of classification (Agreement Level) This parameter was defined in Section 4.2 by using the function $agree(s)$ in Algorithm 1 which corresponds to the number of concordant methods up to 10.

Table 4.4 shows F1 results for each number of agreements. As we have a total of ten base methods, this table shows bootstrapping results when we use instances that 4 or more methods agree with each other, 5 and so on. We did not show results with less than 3 agreements since there were no instances in such scenario.

	#Concordants							
DATASET	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
english_dailabor	69.58	69.18	69.23	68.50	66.91	64.81	59.58	60.75
aisopos_ntua	60.93	60.54	56.87	57.74	64.14	59.65	54.58	58.93
tweet_semevaltest	63.54	63.58	63.64	63.47	63.82	61.56	59.36	59.15
sentistrength_twitter	56.75	58.32	59.13	58.34	57.58	55.35	54.27	57.44
sentistrength_youtube	55.63	55.22	55.67	56.39	56.65	55.44	54.69	54.50
sanders	56.23	55.72	55.94	55.64	55.27	52.73	50.77	48.14
sentistrength_digg	51.13	51.29	53.86	54.58	51.51	50.98	48.06	51.83
sentistrength_myspace	46.76	48.30	48.71	50.31	50.52	51.02	54.34	39.44
sentistrength_rw	48.96	50.09	46.62	49.73	49.00	46.52	46.03	47.58
sentistrength_bbc	49.15	50.62	46.95	47.41	46.31	45.04	45.75	46.57
debate	45.61	45.82	43.69	43.97	44.47	44.99	43.57	43.10
nikolaos_ted	46.29	44.71	48.13	46.43	46.97	47.52	44.50	47.24
vader_nyt	36.13	36.19	36.32	37.35	38.21	36.89	32.54	34.02

Table 4.4. Comparative table of results (F1) for 10SENT bootstrapping by different agreement levels among the base methods in classification

As we can see in this table, the extreme cases of agreement or disagreement produce the worst results. There is a small amount of instance with 100% of agreement, which harms the training of the algorithm. On the other hand, when the agreement is very low, there is a lot of noise in the training data. In sum, the Agreement level represents a trade off between the amount of available data for training and the amount of noise.

The second parameter was the RF confidence Level, defined in Section 4.2 in algorithm 1 as the constant C. The Confidence Level is the confidence ratio of the Random Forest algorithm in its predictions. We use this in order to add more data to train during the bootstrapping step. Then, a similar variation of this parameter was tested, as shown in Table 4.5.

As a final conclusion of these experiments, we arrive at a value of 7 for agreement and 0.7 for confidence, in most datasets, as a way to achieve the "best" balance between quantity and quality for the training data.

Finally, to better understand the aforementioned tradeoff between noise and amount of training data, we ran an experiment in which, instead of using the label in which the methods agreed upon, we use the real labels, with the same instances used in the previous experiment for each agreement level. The results are shown in Table 4.6. In the Table, we can see that for the levels of agreement that produce the best trade-offs (i.e. 6-8)s the results with the real labels are very close to the ones obtained with the agreed label, meaning that the level on noise introduce by this approach is not so high. This also means that majority voting is a reliable way to produce the

	Confidence Level							
DATASET	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
english_dailabor	67.80	66.82	67.28	67.50	67.65	67.63	67.57	67.64
aisopos_ntua	64.53	64.19	63.81	64.95	66.21	60.89	57.57	57.36
tweet_semevaltest	62.56	62.88	62.75	62.65	62.82	63.33	63.21	63.02
sentistrength_twitter	58.11	58.08	58.47	59.24	58.14	56.75	57.71	56.12
sentistrength_youtube	56.64	56.50	55.59	55.55	56.28	56.93	56.86	56.04
sanders	55.22	55.88	54.83	54.62	55.65	54.70	53.81	53.78
sentistrength_myspace	51.86	51.77	52.89	52.46	55.22	52.75	54.51	53.03
sentistrength_digg	51.75	50.98	53.15	51.93	52.59	51.11	50.86	50.85
sentistrength_rw	46.46	45.60	50.24	49.61	50.84	50.59	45.77	47.30
sentistrength_bbc	45.67	45.75	46.16	45.17	47.19	48.00	46.01	48.24
debate	45.77	45.95	46.10	46.53	45.48	45.26	43.94	43.93
nikolaos_ted	45.80	44.70	43.05	46.42	44.76	46.00	45.79	43.48
vader_nyt	38.53	38.33	38.49	38.65	37.69	37.27	36.94	36.10

Table 4.5. Comparative table of results (F1) for 10SENT by different confidence levels added to training in classification.

initial training for our bootstrapping method.

	3	4	5	6	7	8	9	10
english_dailabor	75.20	75.25	74.92	74.18	71.42	70.87	69.95	68.07
aisopos_ntua	75.58	76.13	73.75	73.51	71.61	72.15	70.08	29.06
tweet_ semevaltest	66.68	66.62	66.09	65.66	65.07	64.01	62.54	25.07
sentistrength_ twitter	65.06	65.66	65.45	64.65	64.84	64.01	63.23	61.75
sentistrength_ youtube	61.40	61.54	61.42	61.01	59.62	58.83	57.87	55.33
sanders	61.45	61.27	62.17	60.09	59.36	56.29	53.83	55.78
sentistrength_digg	56.91	56.55	55.15	56.30	53.34	54.46	50.59	45.40
sentistrength_ myspace	56.68	55.83	56.45	57.34	56.03	53.92	54.06	39.07
nikolaos_ted	57.29	57.19	56.55	54.88	48.91	50.22	45.90	18.23
debate	58.62	58.17	57.78	56.20	54.82	55.05	54.60	40.59
sentistrength_rw	52.97	52.88	52.95	51.41	51.65	49.25	47.01	36.12
sentistrength_bbc	43.71	43.33	42.61	41.81	44.30	43.70	37.24	9.56
vader_nyt	45.79	45.22	46.06	46.31	45.95	48.06	47.09	27.89

Table 4.6. Full supervised experiment using with real labels in training phase for each agreement level

4.9 Bag of Words vs. Predictions

After the definitions of the parameters for the classification process, additional features can be extracted and combined with the predictions of other methods to improve results . One example is the text of messages itself. With the text, we can extract the Bag of Words representation of the sentences included in the training.

For each dataset, the Bag of Words was calculated based on the occurrence of each word in the text of the instance compared with corpus of all dataset. This was concatenated with the results of each method to the classification. In Table 4.7 we can find the results that compare the use of these different sets of features in 10SENT, the predictions outputted by all base methods and bag of words. Here, we used all best parameters discovered in previous sections, including the random forest classifier for the learning step of algorithm. We used a fixed number of trees as parameter in RF for all experiments, because after certain amount of trees it "stabilizes" a bit, so more trees could be better, but there is not much variation in terms of results.

Note that the combination of these two set of features improves results compared with each single one separately. Despite Bag of Words individually presented better results in a few datasets it is not the best in all of them and alone, which suggests that using both features is the best option for 10SENT. In the next experiments, we always use this joint representation (BoW + BaseMethods), when me mention 10SENT.

Dataset	Bag of Words	BaseMethods	BoW + BaseMethods
english_dailabor	68.4	67.1	72.4
aisopos_ntua	72.3	62.0	69.9
tweet_semevaltest	58.3	62.8	65.2
sentistrength_twitter	58.8	59.1	61.2
sentistrength_youtube	56.6	56.1	58.7
sanders	61.5	54.1	56.4
sentistrength_myspace	50.2	52.3	52.2
sentistrength_digg	45.4	50.1	50.6
nikolaos_ted	51.3	45.9	49.0
debate	57.1	45.9	47.1
sentistrength_rw	48.3	48.5	45.5
sentistrength_bbc	34.8	45.5	43.8
vader_nyt	28.0	38.9	39.2

Table 4.7. Results of 10SENT using different set of features for classifier in Random Forest

Chapter 5

Transfer Learning and Active Learning

Next, we present some other fields of research which we explore to improve the results of 10SENT: transfer learning and active learning. The following techniques seek to use the learning and knowledge contained in the data from other sources in order to add extra information into the algorithm in a smart way.

5.1 Active Learning: Using ALAC

For active learning, we analyze a situation in which a user, wanting to apply our proposed techniques, is willing to provide "a little bit of help" to our system by labeling a few, but potentially very informative, instances to the classifier training process.

To perform such analysis, we rely on the use of a state-of-the-art active learning method. Active learning methods act in a phase prior to the actual instances labeling to create a training set for a classifier. Their goal is to select the most informative and diverse set of instances from an unlabeled dataset that can maximize the learning process of the classifier while minimizing the labeling effort, i.e., the selected set should be as small as possible.

We have chosen for this analysis the ALAC (standing for Active Learning Associative Classifier) [Silva et al., 2011], an active learning method based on association rules. When compared to other active learning methods reported in the literature, ALAC has several advantages such as: (1) it does not require an initial labeled set, unlikely approaches based on committees; (2) it has a clear stopping criterion, a property that many approaches do not possess; and (3) it can select very few but highly

informative instances based on an informativeness criteria grounded on lazy association rules [Silva et al., 2014].

In particular, we analyze the application of two variants of ALAC: (1) the first one (called StandardALAC) corresponds to the original method as designed, with its stop criterion; and (2), in the second one (called ALAC10%), we "turned off" the stop criterion and let the method rank the "best" instances to be labeled, based on its heuristics, until achieving around 10% of the unlabeled set available for training.

The results of those tests with active learning are shown in Table 5.1. In this table, we compare the use of both forms of ALAC with 10SENT and also we performed an experiment using only ALAC as source of training data, so we can observe the importance of maintaining 10SENT label prediction as criteria to the method.

	10sent	10Sent + ALAC10%	10Sent + ALACstandard	ALAC
english_dailabor	70.62	70.94	72.37	68.05
aisopos_ntua	69.91	69.34	70.10	55.87
tweet_semevaltest	64.78	64.31	65.37	56.54
sentistrength_twitter	62.17	62.79	62.14	49.03
sentistrength_youtube	57.06	58.42	58.37	49.66
sanders	56.19	58.26	56.57	48.19
sentistrength_digg	51.91	51.66	51.00	18.88
sentistrength_myspace	50.22	51.14	50.18	43.70
nikolaos_ted	47.97	48.17	47.48	30.39
debate	47.18	49.31	47.37	47.67
sentistrength_rw	47.15	47.94	45.54	44.06
sentistrength_bbc	43.76	42.18	44.86	26.28
vader_nyt	39.81	41.89	39.26	26.09

Table 5.1. Macro-F1 results for each experiment on 10SENT using ALAC as Active Learning

5.2 Transfer Learning Analysis

Finally, we evaluate whether it is possible to explore some "easily available" knowledge from an external source. We do this by exploring datasets in which messages are labeled with "emoticons" by the systems users themselves.

In a task of machine learning, the transfer learning occurs when an algorithm uses knowledge obtained while solving a specific problem (source-task) and applying it to a different but related one (target-task). But inductive transfer can be viewed not only as a way to improve learning in a standard supervised-learning task, but also

Label	Emoticon
Positive	:) :] :} :o) :o] :o} :-] :-) :-} =) =] =} =^] =^) =^} :B :-D :-B :^D :^B =B =^B =^D :') :'] :'} =') ='] ='} <3 ^.^ ^-^ ^_^ ^^ :* =* :-* ;) ;] ;} :-p :-P :-b :^p :^P :^b =P =p \o\ \o/ /o/ :P :p :b =b =^p =^P =^b
Negative	D: D= D-: D^: D^= :(:[:{ :o(:o[:^(:^[:^{ =^(=^{ >=(>=[>={ >=(>:-{ >:-[>:-(>=^[>:-(:-[:-(=(=[={ =^[>:-=(>=[>=^(:'(:'[:'{ ='{ ='(='[=\ :\ =/ :/ =$ o.O O_o Oo :$:-{ >:-{ >=^{ :o
Neutral	:| =| :-| >.< >< >_< :o :0 =O :@ =@ :^o :^@ -.- -.-' -_- -_-' :x =X :# =# :-x :-@ :-# :^x :^#

Table 5.2. List of Emoticons divided by categories

as a way to offset the difficulties posed by tasks that involve unsupervised learning, semi-supervised learning, or small datasets [Torrey and Shavlik, 2009].

That is, if there are small amounts of data or class labels for a task, treating it as a target task and performing inductive transfer from a related source task can lead to more accurate models.

To use an approach that transfers knowledge from one task to another, it is usually necessary to map characteristics from the source problem into the destination problem, identifying similarities and differences. Next we detail how we transfer knowledge from existing emoticons in the datasets to the task of sentence-level sentiment analysis.

5.2.1 Mapping Emoticons to Sentiment Analysis

Emoticons became very popular nowadays and even the English Dictionary of Oxford has recently chosen an "emoji" as word of year (in 2015) due to its notable and massive use around world [Dictionaries, 2015]. Emoticons are representations of an expression in a faced-look set of characters, although there is a wide range of non-facial variations. They are used to give us an idea of feelings in the text, like happiness or sadness.

Previous work has demonstrated that such messages, though not available in large volumes, are very precise. In other words, labeling with emoticons used by the final user indeed provide trustful information about the polarity of message. Accordingly, in these experiments, we used the "rules of thumb" suggested in [Gonçalves et al., 2013] to translate emoticons into polarities.

To map these characteristics in our sentiment analysis task, we have to divide them in groups, matching a large set of common emoticons with our previous labels (positive, negative and neutral). First, we identify a set of emoticons, then we separate them in three classes as listed in Table 5.2. This table also includes the popular variations that express the primary polarities of positive, negative, and neutral.

	Accuracy	Coverage
nikolaos_ted	0.919	0.014
sentistrength_myspace	0.800	0.091
aisopos_ntua	0.787	0.526
tweet_semevaltest	0.693	0.071
english_dailabor	0.687	0.064
sentistrength_youtube	0.686	0.085
sentistrength_twitter	0.627	0.097
sentistrength_rw	0.619	0.148
sentistrength_digg	0.600	0.028
sanders	0.359	0.045
debate	0.339	0.015
sentistrength_bbc	0.173	0.006
vader_nyt	-	-

Table 5.3. Accuracy and coverage of emoticons in training experiments for all datasets

As one may expect, the fraction of messages containing at least one emoticon is very low compared to the total number of messages that could express emotion. A recent work has identified that this rate is less than 10% [Park et al., 2013]. As we can see at Table 5.3 emoticons appeared just in a very small amount of instances, which we can observe by the coverage column. Despite of that, the accuracy of emoticons is often very precise to distinguish polarity of sentiment, reaching more than 90% in nikolaos_ted dataset. This is also in agreement with previous efforts [Gonçalves et al., 2013].

Our ultimate goal here is to extract some information about the text of those messages to our classification step. For this, we incorporate into the training data these instances labeled with emoticons extracted from datasets.

Messages with more than one emoticon were associated with the polarity of the first emoticon that appeared in the text, although we encountered only a small number of such cases in the data.

To compare the effect of transfer learning from emoticons, we separated it in three different experiments: first with our traditional 10SENT; after we used just emoticon labels to create training, without our majority predictions; then we combined these two to check the impact of emoticons in our method. Results of this experiment can be seen in Table 5.4. We can see that improvements of up to 6% (e.g., in case of the sentistrength_myspace dataset) can be obtained in terms of Macro F1, with no significant losses in most datasets and with no extra (labeling) cost.

For comparative results, we included another table with the results for the Major-

	10Sent	Emoticons	10Sent + Emoticons
english_dailabor	70.62	25.57	72.02
aisopos_ntua	69.91	35.48	73.61
tweet_semevaltest	64.78	18.06	65.13
sentistrength_twitter	62.17	22.93	62.87
sentistrength_youtube	57.06	-	59.36
sanders	56.19	11.83	56.78
sentistrength_digg	51.91	-	52.22
sentistrength_myspace	50.22	-	53.20
nikolaos_ted	47.97	-	48.97
debate	47.18	-	47.37
sentistrength_rw	47.15	-	45.25
sentistrength_bbc	43.76	-	43.18
vader_nyt	39.81	-	39.01

Table 5.4. Macro-F1 results for experiments on 10SENT using Transfer Learning

ity Voting and Best Individual methods included. We highlight that these comparisons should be analyzed with some caution as Majority Voting is an unsupervised basic way to combine methods but choosing the "best" base method for a given dataset without supervision (i.e., labeling) is an open research issue and, thus, it does not represent a realistic or practical baseline scenario. We can see from the Table that a "bit of help" from the user in labeling instances can indeed improve results. In general, gains over the version without any labeled supervision vary from 2.4% to 5.2%. However, based on the results reported in the Table, there is not a clear winner between the two approaches. Notice that training the classifier with only the labeled dataset chosen by ALAC (column ALAC10% in Table 5.5) produces poor results, meaning that our heuristics are very important to produce good results.

Overall, there is no statistically significant losses between the approaches, based on t-test with 95% confidence level, but there are clear improvements in some datasets. Thus, this approach represents an interesting opportunity to provide to the user some help in terms of labeling effort.

It is important to highlight that a transfer learning method aims at producing *positive transfer* between tasks and avoiding *negative transfer*. By using emoticons, we can see that datasets with higher incidence of emoticons tends to lead to a higher positive transfer. On the other hand, a small amount of emoticons is not correlated with a negative transfer. We speculate that negative transfer might happen in datasets that contains more cases of sarcasms and irony. But we let this topic to be further investigated as part of our future work.

	10SENT	10SENT + Alac10%	10SENT + Standard Alac	10SENT + Emoticons	ALAC10%	Majority Voting	Best Individual
english_dailabor	70.62	70.94	72.37	72.02	68.05	59.45	74.58
aisopos_ntua	69.91	69.34	70.10	73.61	55.87	68.16	67.47
tweet_semevaltest	64.78	64.31	65.37	65.13	56.54	62.64	61.27
sentistrength_twitter	62.17	62.79	62.14	62.87	49.03	58.89	59.05
sentistrength_youtube	57.06	58.42	58.37	59.36	49.66	54.60	56.81
sanders	56.19	58.26	56.57	56.78	48.19	54.75	53.52
sentistrength_digg	51.91	51.66	51.00	52.22	18.88	51.50	51.98
sentistrength_myspace	50.22	51.14	50.18	53.20	43.70	51.56	54.05
nikolaos_ted	47.97	48.17	47.48	48.97	30.39	47.17	50.76
debate	47.18	49.31	47.37	47.37	47.67	48.34	47.97
sentistrength_rw	47.15	47.94	45.54	45.25	44.06	43.99	46.45
sentistrength_bbc	43.76	42.18	44.86	43.18	26.28	45.19	46.17
vader_nyt	39.81	41.89	39.26	39.01	26.09	37.19	44.56

Table 5.5. Macro-F1 results for each experiment on 10SENT compared with other methods

Chapter 6

Comparative Results

We now turn our attention to the comparison between 10SENT, the "strongest" baseline (Majority voting) and the base methods. We should point out that in these comparisons, and in all others in the Chapter, a mention to 10SENT corresponds to the results obtained with the best unsupervised configuration found in the previous analyses, in other words, the original 10SENT representation (methods' decisions) along with the BagOfWords and the transfer learning.

We can observe in Figure 6.1 that our method has a higher Macro-F1, above the baselines, in most datasets. In fact, 10SENT is the best method in 7 out of 13 datasets and it is close to the top of the rank in several others. This is also reflected in the Mean Rank, shown in Table 6.1, confirming that 10SENT is the overall winner across all tested datasets.

METHOD	**MEAN RANK**	**POS**	**DEVIATION**
10SENT	2.154	1	1.457
Majority Voting	3.154	2	1.350
Vader	3.692	3	1.814
SO-CAL	3.769	4	1.717
Umigon	4.923	5	2.921
Afinn	6.615	6	1.820
OpinionLexicon	6.923	7	1.900
pattern.en	7.000	8	2.287
OpinionFinder	9.308	9	1.136
Sentistrength	9.846	10	1.747
Emolex	9.923	11	2.055
Sentiment140 Lexicon	10.692	12	2.493

Table 6.1. Mean Rank of methods for all datasets

In fact, 10SENT can be considered as the most *stable method* as it produces the

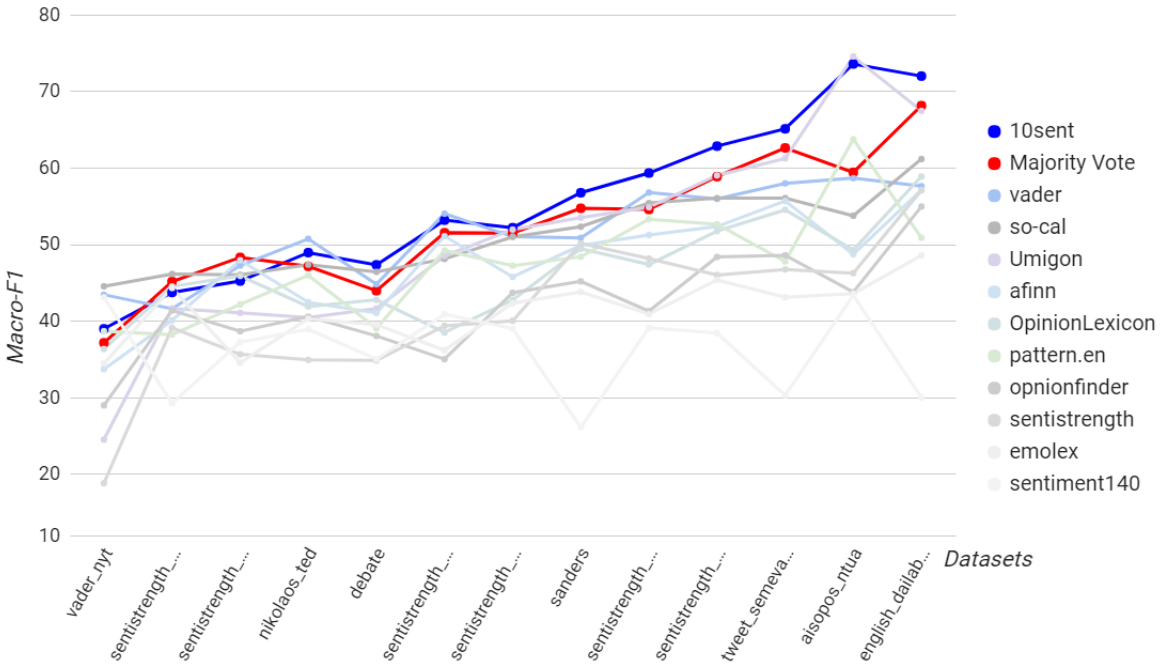

Figure 6.1. Macro-F1 results of 10SENT compared with each individual base method for all datasets

best (or close to the best) results in most datasets in different domains and applications. In other words, by using our proposed method, one can almost always guarantee top-notch results, at no extra cost, and without the need to discover the best method for a given context/dataset/domain. Figure 6.2 shows the rank of 10SENT for all datasets, which demonstrate how its position does not vary much between datasets compared to other methods.

6.1 UpperBound Comparison

For analysis purposes, we perform a comparison of our unsupervised method with some "uppperbound" baselines which use some type of privileged information, most notably the real label of the instances in the training set, an information not available for 10SENT. The idea here it to understand how far our proposed unsupervised approach is from the ones that exploit such information as well as to understand the limits to what we can achieve with an unsupervised one.

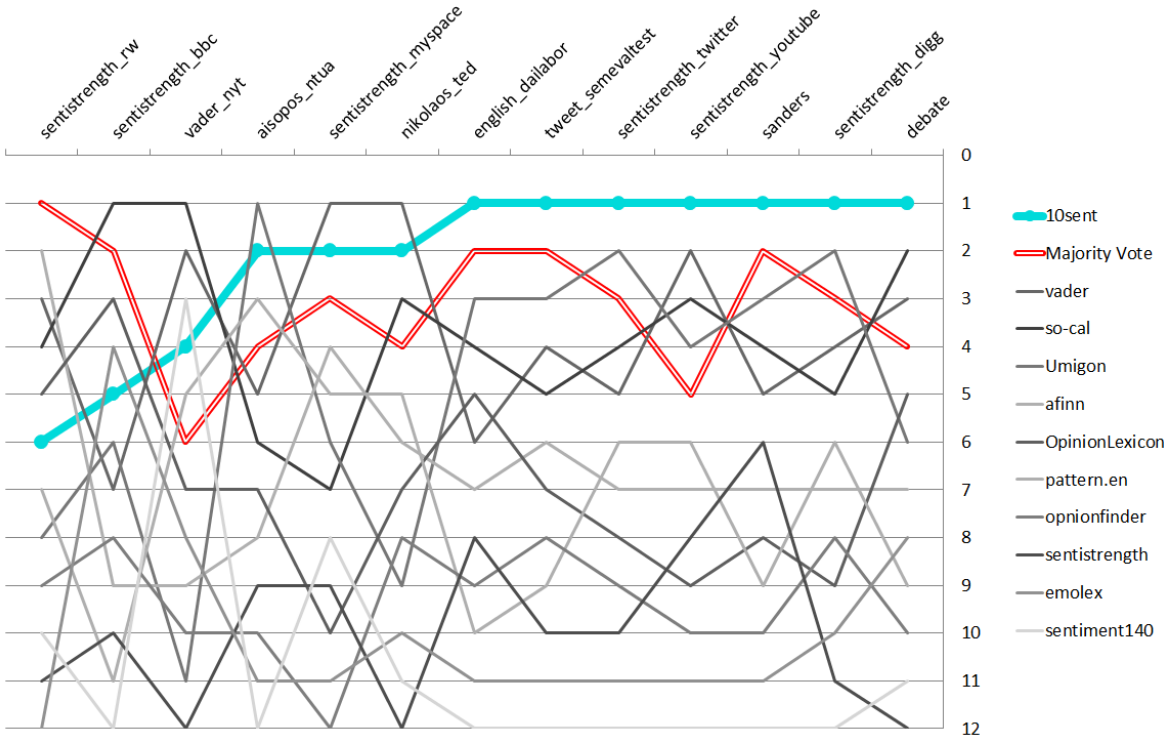

Figure 6.2. Mean Rank of 10SENT compared with other methods for all datasets

6.1.1 Fully Supervised

The first "upperbound" baseline is a fully supervised approach which uses all the labeled information available in the training data of the 5-fold cross-validation procedure. As a normally done in fully supervised approaches, the parameters of the RF algorithm are determined using a validation set (aka, nested cross-validation within the training set). As we are developing an unsupervised method, a fully supervised experiment is considered here as a upperbound.

6.1.2 Exhaustive Weighted Majority Voting

The second baseline is an *Exhaustive Weighted Majority Voting* method that uses the real labels of messages of the datasets to find the best possible linear combination of weights for each base method. Differently from the Majority Voting baseline, in which all methods have the same weight, in this approach, each individual base method has a different weight, so that the influence of each one in the final classification is different. It is worth mentioning that weights here are not fixed. Each dataset has its own weights calculate isolatedly.

As explained before, the search for the weights was performed in exhaustive mode, in other words, we evaluate every possible combination, seeking to maximize the Macro-F1 in each dataset. During the experiments, we limited the search to five different weights in the range $[0 - 1]$: $W = \{0, 0.25, 0.5, 0, 75, 1\}$ to estimate "close-to-best" results, while maintaining feasible computational costs.

Table 3.5 shows the average weights and corresponding standard deviation for each method across all datasets. We can see that most methods have different behaviors in different datasets (implied by the large deviations). The same method may have a huge variance in effectiveness in different datasets, which precludes the use of a single unique set of weights for all cases. Despite this, we can observe that some methods have clearly a higher average than others even with this high deviation.

6.1.3 Best Individual Method

Finally, the third "upperbound" baseline is the best single base method in each dataset. Since the base methods are unsupervised "off-the-shelf" ones, we determine the best method for each dataset also using the labels in the training sets. It is also an "upperbound" because the best method to a dataset cannot be determined, in advance, without supervision, i.e., a manually labeled training set. We call *Best Individual* the results composed by the result of this "best_method" for each dataset

6.1.4 Upperbound Results

The results of those upperbounds are shown at Table 6.2. For comparative purposes we also included in this table the results of the unsupervised Majority Voting. As before, all results correspond to the average performance in the 5 test sets of the folded cross-validation procedure using 10SENT with its best configuration including Bag of Words and Transfer Learning.

Values marked with "*****" in this table indicate that the difference was not statistically significant when compared to the 10SENT in a paired-test with 95% confidence; results reported with "$^{\triangle}$" are those statistically better than those of 10SENT. On the other hand, our method demonstrated to be statistically superior to the ones whose values are marked with "$^{\nabla}$";

As highlighted before, 10SENT is tied or better than the traditional majority voting in most datasets, being statistically superior seven out of 12 cases, tying in other 5 and losing only in one dataset (sentistrength_rw). Gains can achieve up to 23.8% against this baseline. When compared to the best individual method in each

	Fully Supervised	Exhaustive Weighted Majority Voting	Best Individual	Majority Voting	10SENT
aisopos_ntua	$76.64^{\triangle}$	$75.8^{\triangle}$	74.58*	$59.45^{\triangledown}$	73.61
english_dailabor	$75.63^{\triangle}$	71.9*	$67.47^{\triangledown}$	$68.16^{\triangledown}$	72.02
tweet_semevaltest	66.77*	65.5*	$61.27^{\triangledown}$	$62.64^{\triangledown}$	65.13
sentistrength_twitter	$66.14^{\triangle}$	62.9*	$59.05^{\triangledown}$	$58.89^{\triangledown}$	62.87
sentistrength_youtube	$61.77^{\triangle}$	60.6*	$56.81^{\triangledown}$	$54.60^{\triangledown}$	59.36
sanders	$62.76^{\triangle}$	$58.0^{\triangle}$	53.52*	54.75*	56.78
sentistrength_myspace	$57.47^{\triangle}$	$57.8^{\triangle}$	54.05*	51.56*	53.20
sentistrength_digg	$59.52^{\triangle}$	$57.3^{\triangle}$	51.98*	51.50*	52.22
nikolaos_ted	$57.43^{\triangle}$	$56.1^{\triangle}$	$50.76^{\triangle}$	47.17*	48.97
debate	$58.75^{\triangle}$	$49.1^{\triangle}$	46.45*	$43.99^{\triangledown}$	47.37
sentistrength_rw	$53.52^{\triangle}$	$52.2^{\triangle}$	47.97*	$48.34^{\triangle}$	45.25
sentistrength_bbc	44.00*	$51.8^{\triangle}$	46.17*	45.19*	43.18
vader_nyt	$46.87^{\triangle}$	$51.9^{\triangle}$	$44.56^{\triangle}$	$37.19^{\triangledown}$	39.01

Table 6.2. Results in terms of Macro-F1 comparing 10SENT with all other evaluation methods ("*" indicates values that the difference was not statistically significant compared to the 10SENT; "$^{\triangledown}$" are values that 10SENT wins and "$^{\triangle}$" are the values statistically superior to the 10SENT result)

dataset, 10SENT wins (4 cases) or ties (7 cases) in 11 out of 13 cases, a strong result. This shows that 10SENT is a good and consistent choice among all the method options that are available here, independently of which dataset is used.

When compared to supervised Exhaustive Weighted Majority Voting, a first observation is that, as expected, it is always superior to the simple Majority Voting. Although we cannot beat this "upperbound" baseline, we tie with it in 4 datasets (sentistrenth_youtube, sentistrength_twitter, tweet_semevaltest, english_dailabor) and get close results in others such as aisopos_ntua, sanders and debate. This with no cost at all in terms of labeling effort.

Regarding the strongest upperbound baseline – Fully Supervised –, an interesting observation to make is that in some datasets its results get very close to those of the Exhaustive Weighted Majority Voting in several datasets, even lossing to it in two (sentistrength_bbc, vader_nyt). This is a surprising result, meaning that the combination of both strategies is also an interesting venue to pursue in the future. When comparing this baseline to 10Sent, as expected, we can also not beat it, but can tie with it in two datasets and get close results in others, mainly in those cases in which our method was a good competitor against Exhaustive Weighted Majority Voting. We consider these very strong results.

To a deeper understanding of these results, Table 6.3 shows the set size of 10SENT used to train the classifier in the first portion of our algorithm (Lines 5-10 of Algorithm 1), chosen based on the majority voting, and the accuracy of the automatic labeling in the training, before and after the bootstrapping step (lines 9-11 of Algorithm 1).

	10SENT					
	Majority Voting			Bootstrapping		
	Set Size	Accuracy	F1	Set Size	Accuracy	F1
english_dailabor	1999	0.858	70.61	2165	0.826	70.62
aisopos_ntua	215	0.762	71.67	238	0.734	69.91
tweet_semevaltest	2781	0.796	65.04	3139	0.757	64.78
sentistrength_twitter	2042	0.706	63.06	2238	0.665	62.17
sentistrength_youtube	1688	0.660	58.37	1837	0.645	57.06
sanders	1760	0.762	55.94	1929	0.758	56.19
sentistrength_digg	474	0.630	49.32	519	0.615	51.91
sentistrength_myspace	488	0.641	49.34	535	0.645	50.22
debate	1422	0.508	47.30	1620	0.530	47.97
nikolaos_ted	329	0.606	47.11	370	0.556	47.18
sentistrength_rw	417	0.618	43.12	471	0.601	47.15
sentistrength_bbc	376	0.687	37.91	418	0.661	43.76
vader_nyt	2222	0.363	40.44	2563	0.366	39.81

Table 6.3. Set size, "noise"(indicated by accuracy) and Macro-F1 values to 10SENT training sets without bootstrapping and including bootstrapping step

As we can see the majority voting heuristics (original training set) selects a relative large amount of training data from the original unlabeled datasets – in average, around 50% of the unlabeled datasets available for training are selected based on the chosen parameters. This may explain some of the good results obtained in our experiments, since the classifiers have a reasonable amount of data to be trained with. We can also see that the bootstrapping step is also capable of increasing the size of these training sets (column "Bootstrapping - Set Size)" in about 10% in average considering all datasets. This also explains some of the improvements obtained by this step. In fact, when comparing the column "F1" of Table 6.3 in the case of "Majority Voting" with the Bootstrapping column, we can see that this step never produces (statistically significant) losses in any dataset, but can indeed produce large improvements in Macro F1, such as in the case of the sentistrength_rw and sentistrength_bbc datasets, with gains of up to 15%.

Our results show that the task of detection of polarity for sentiment analysis still have a space to be further explored, since the highest Macro F1 found is around 75%. Although, some works demonstrate that even human labeling could be also problematic. In Thelwall [2013]. for example, they conclude that texts with presence of sarcasm or irony are difficult to be labeled, perhaps because its power is partly due to the cleverness with which it is constructed. Pang and Lee [2005] also examined

human accuracy at determining relative positivity in reviews for sentiment analysis and, though they conclude human can discern evaluation scores, there is a large variation in accuracy between subjects. Therefore, even humans do not reach the highest level of accuracy, then we cannot expect an algorithm to do so.

However, this is only part of the story. One question that remains to be answered is: "What is the **quality** of the automatically labeled training set". We can answer this question by looking at the columns "Accuracy" in the original training set and after the bootstrapping step. This metric calculates the proportion of correctly assigned labels in the training sets when compared to the "real labels". We can see that for a considerable number of datasets the accuracy is relatively high, between 0.6-0.8. In fact, the cases in which 10SENT gets closer to the fully supervised method correspond to those in which the accuracy in the training is higher. We can also see that after the bootstrapping, in general the accuracy in the training drops a bit, which is natural since the heuristics based on classifier confidence is not perfect, but this is compensated by the increase in training size, resulting in a learned model that generalizes better.

Finally, we can see that the absolute results of the best overall method in each dataset are still not very high (maximum of 76%), which shows the difficulty of the sentiment analysis task and that there is a lot of room for improvements.

Chapter 7

Conclusion

7.1 Concluding Discussion

We present a novel unsupervised approach for sentiment analysis on sentence-level derived from the combination of several existing "off-the-shelf" sentiment analysis methods. Our solution was thoroughly tested in a wide and diversified environment. We cover a vast amount of methods and labeled datasets from different domains. Next, we briefly discuss the main benefits of our proposed method and we discuss how we tackle its possible limitations.

Unsupervised adaptation to the data context: The key advantage of our combined approach in the context of sentence-level sentiment analysis is that it fixes the key issue in this field. Recent efforts by Ribeiro et al. [2016] have shown that the prediction performance of popular unsupervised approaches varies considerably from one dataset to another. As many researchers are just interested in using a valid method and the state-of-the-art has not been clearly established, researchers tend to accept any popular method as a valid methodology. It is common to see concurrent important papers, sometimes published in the same venue, using completely different methods without any justification for the choice.

Our experimental results show that our approach has the lowest prediction performance variability. Our self-learning approach provides to this method the ability to slight adapt itself to different contexts, maintaining good prediction performance for data from different contexts. This strategy is smarter than creating a combined lexicon as our approach can give higher value to the votes of one existing method over the other, providing some level of adaptability to a still unsupervised approach. This is a key issue in an area in which researchers are mostly interested in using an "off-the-shelf" method to different contexts. Moreover, our approach is easily expandable to

include any new developed unsupervised method. Furthermore, by the structure of the combination, our method is not influenced by variations in vocabulary size or domain idiosyncrasies.

Improving the state-of-the-art performance: Our experimental results show that 10SENT achieves good effectiveness compared to our baselines in analyzing sentiment expressed in messages of many different social media data sources. 10SENT showed to be better than all existing individual methods and also obtained better results than the traditional majority voting strategy, with gains of up to 17.5%. In an upper bound comparison, we could also see that 10SENT can get close to the best supervised results in several situations, meaning that our approach leaves only a limited space for future improvements. On the other hand, our analysis on transfer learning shows us the possibility of adapting the method to include more strategies that can lead to better results.

Tackling the additional combination complexity: A common criticism that usually relies upon combined approaches is that they lead to more complex systems, making them often hard to be used in practice. As an attempt to fix the additional complexity of combining results of many methods we will release our codes and datasets to the research community. To make our method easy to use, we also plan to deploy it as a free online API. We hope that, by making it easy to use, the added complexity should not be a barrier for the use of our method.

7.2 Future Work

We envision a few directions that can be explored as future directions of this study. We aim at better exploring weights as well choosing other setup of methods for different scenarios of data. Additionally, it is possible to explore more the corpus of the text dataset and identify features to improve classification.

Other path we aim to follow is that the selection of "*top 10 methods*" could be deeper explored. We can analyze intrinsic correlation between methods, not only ten, but a vast collection of tools we have at hand in literature for sentiment analysis and select informative and distinct ones that could provide a better classification of sentiment polarity in a text.

We also plan to deploy our method as part of known sentiment analysis benchmark systems Araújo et al. [2016].

Bibliography

(2005). Amazon mechanical turk. `https://www.mturk.com/`. Accessed June 17, 2013.

Araújo, M., Gonçalves, P., Benevenuto, F., and Cha, M. (2014). ifeel: A system that compares and combines sentiment analysis methods. In *World Wide Web Conference (Companion Volume)*.

Araújo, M., Diniz, J. P., Bastos, L., Soares, E., Júnior, M., Ferreira, M., Ribeiro, F., and Benevenuto, F. (2016). ifeel 2.0: A multilingual benchmarking system for sentence-level sentiment analysis. In *Proceedings of the International AAAI Conference on Web-Blogs and Social Media*, Cologne, Germany.

Bollen, J., Mao, H., and Zeng, X.-J. (2010). Twitter Mood Predicts the Stock Market. *CoRR*, abs/1010.3003.

Bradley, M. M. and Lang, P. J. (1999). Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical Report C-1, The Center for Research in Psychophysiology, University of Florida.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5--32.

Cambria, E., Speer, R., Havasi, C., and Hussain, A. (2010). Senticnet: A publicly available semantic resource for opinion mining. In *AAAI fall symposium: commonsense knowledge*, volume 10, page 02.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1--27:27. ISSN 2157-6904.

Chaudhuri, A. (2006). *Emotion and reason in consumer behavior*. Routledge.

Dang, Y., Zhang, Y., and Chen, H. (2010). A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. *IEEE Intelligent Systems*, 25(4):46–53.

Dictionaries, O. (2015). Oxford dictionaries word of the year 2015 is.... http://blog.oxforddictionaries.com/2015/11/word-of-the-year-2015-emoji/. Accessed: 2017-04-30.

Dietterich, T. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg.

Dodds, P. S. and Danforth, C. M. (2010). Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. *Journal of Happiness Studies*, 11(4):441--456.

Douglas, J. (1978). Chess 4.7 versus david levy: The computer beats a chess master. *Byte, December*, 1978:78--90.

El-Halees, A. et al. (2011). Arabic opinion mining using combined classification approach.

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417--422. Citeseer.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *Processing*, -:1--6.

Gonçalves, P., Araújo, M., Benevenuto, F., and Cha, M. (2013). Comparing and combining sentiment analysis methods. In *Proceedings of the first ACM conference on Online social networks*, pages 27--38. ACM.

Gonçalves, P., Benevenuto, F., and Cha, M. (2013). PANAS-t: A Pychometric Scale for Measuring Sentiments on Twitter. abs/1308.1857v1.

Gonçalves, P., Dalip, D. H., Costa, H., Gonçalves, M. A., and Benevenuto, F. (2016). On the combination of "off-the-shelf" sentiment analysis methods. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, pages 1158--1165. ACM.

Hannak, A., Anderson, E., Barrett, L. F., Lehmann, S., Mislove, A., and Riedewald, M. (2012). Tweetin'in the rain: Exploring societal-scale effects of weather on mood. In *ICWSM*.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. Proc. KDD'04, pages 168--177.

Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3--24, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Levallois, C. (2013). Umigon: sentiment analysis for tweets based on terms lists and heuristics. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 414--417.

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

Messias, J., Diniz, J. P., Soares, E., Ferreira, M., Araujo, M., Bastos, L., Miranda, M., and Benevenuto, F. (2016). Towards sentiment analysis for mobile devices. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '16.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39--41.

Mohammad, S., Kiritchenko, S., and 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 (SemEval-2013)*, Atlanta, Georgia, USA.

Mohammad, S. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Moraes, F., Vasconcelos, M. A., Prado, P., Dalip, D. H., Almeida, J. M., and Gonçalves, M. A. (2013). Polarity detection of foursquare tips. Int'l Conf. on Social Informatics (SOCINFO).

Mudinas, A., Zhang, D., and Levene, M. (2012). Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Int'l Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM)*.

Nielsen, F. Å. (2011). A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Oliveira, N., Cortez, P., and Areal, N. (2013). On the predictability of stock market behavior using stocktwits sentiment and posting volume. In *Proc. EPIA'13*, pages 355–365.

Pang, B. and 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*, pages 115--124. Association for Computational Linguistics.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1--135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79--86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Park, J., Barash, V., Fink, C., and Cha, M. (2013). Emoticon style: Interpreting differences in emoticons across cultures. In *ICWSM*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825--2830.

Prabowo, R. and Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143--157.

Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., and Benevenuto, F. (2016). Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1--29.

Silva, N. R., Lima, D., and Barros, F. (2012). Sapair: Um processo de análise de sentimento no nível de característica. In *4nd International Workshop on Web and Text Intelligence (WTI'12), Curitiba*, page 2.

Silva, R. M., Gonçalves, M. A., and Veloso, A. (2011). Rule-based active sampling for learning to rank. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III*, pages 240--255.

Silva, R. M., Gonçalves, M. A., and Veloso, A. (2014). A two-stage active learning method for learning to rank. *JASIST*, 65(1):109--128.

Smedt, T. D. and Daelemans, W. (2012). Pattern for python. *Journal of Machine Learning Research*, 13(Jun):2063--2067.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254--263. Association for Computational Linguistics.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267--307.

Tausczik, Y. R. and Pennebaker, J. W. (2010). The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24--54.

Thelwall, M. (2013). Heart and soul: Sentiment strength detection in the social web with sentistrength. *Proceedings of the CyberEmotions*, pages 1--14.

Torrey, L. and Shavlik, J. (2009). Transfer learning.

Tsytsarau, M. and Palpanas, T. (2012). Survey on mining subjective data on the web. *Data Min. Knowl. Discov.*, 24(3):478--514. ISSN 1384-5810.

Turney, P. D. and Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315--346.

Wang, H., Can, D., Kazemzadeh, A., Bar, F., and Narayanan, S. (2012). A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In *ACL System Demonstrations*, pages 115--120.

Watson, D., Clark, L. A., and Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, 54(6):1063.

Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). Opinionfinder: A system for subjectivity analysis. In *Proceedings of hlt/emnlp on interactive demonstrations*, pages 34--35. Association for Computational Linguistics.

Yeung, D. S., Liu, Z.-Q., Wang, X.-Z., and Yan, H. (2006). *Advances in Machine Learning and Cybernetics: 4th International Conference, ICMLC 2005, Guangzhou, China, August 18-21, 2005, Revised Selected Papers (Lecture ... / Lecture Notes in Artificial Intelligence)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 3540335846.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. (2011). Combining lexicon-based and learning-based methods for twitter sentiment analysis. Technical report, HP. `http://www.hpl.hp.com/techreports/2011/HPL-2011-89.html`.

Zhou, L. and Chaovalit, P. (2008). Ontology-supported polarity mining. *Journal of the Association for Information Science and Technology*, 59(1):98--110.