Semantic Sentiment Analysis of Microblogs

Thesis

SEMANTIC SENTIMENT ANALYSIS OF MICROBLOGS

HASSAN SAIF

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD)

Knowledge Media Institute
Open University

January 21, 2015

SUPERVISORS:

Dr. Harith Alani

Dr. Miriam Fernandez

Dr. Yulan He

LOCATION:

Milton Keynes, United Kingdom

Dedicated to my beloved mother Nebal Saif.

ABSTRACT

Microblogs and social media platforms are now considered among the most popular forms of online communication. Through a platform like Twitter, much information reflecting people's opinions and attitudes is published and shared among users on a daily basis. This has recently brought great opportunities to companies interested in tracking and monitoring the reputation of their brands and businesses, and to policy makers and politicians to support their assessment of public opinions about their policies or political issues.

A wide range of approaches to sentiment analysis on Twitter, and other similar microblogging platforms, have been recently built. Most of these approaches rely mainly on the presence of affect words or syntactic structures that explicitly and unambiguously reflect sentiment (e.g., "*great*", "*terrible*"). However, these approaches are semantically weak, that is, they do not account for the semantics of words when detecting their sentiment in text. This is problematic since the sentiment of words, in many cases, is associated with their semantics, either along the context they occur within (e.g., "*great*" is negative in the context "*pain*") or the conceptual meaning associated with the words (e.g., "*Ebola*" is negative when its associated semantic concept is "*Virus*").

This thesis investigates the role of words' semantics in sentiment analysis of microblogs, aiming mainly at addressing the above problem. In particular, Twitter is used as a case study of microblogging platforms to investigate whether capturing the sentiment of words with respect to their semantics leads to more accurate sentiment analysis models on Twitter. To this end, several approaches are proposed in this thesis for extracting and incorporating two types of word semantics for sentiment analysis: contextual semantics (i.e., semantics captured from words' co-occurrences) and conceptual semantics (i.e., semantics extracted from external knowledge sources).

Experiments are conducted with both types of semantics by assessing their impact in three popular sentiment analysis tasks on Twitter; entity-level sentiment analysis, tweet-level sentiment analysis and context-sensitive sentiment lexicon adaptation. Evaluation under each sentiment analysis task includes several sentiment lexicons, and up to 9 Twitter datasets of different characteristics, as well as comparing against several state-of-the-art sentiment analysis approaches widely used in the literature.

The findings from this body of work demonstrate the value of using semantics in sentiment analysis on Twitter. The proposed approaches, which consider words' semantics for sentiment analysis at both, entity and tweet levels, surpass non-semantic approaches in most datasets.

ACKNOWLEDGEMENTS

When I think of my PhD experience over the past three and half years, two things come to my mind. Firstly, my PhD was an exciting journey that reshaped my life entirely. Secondly, I have been amazingly fortunate to meet people who have walked with me through this journey step by step and inch by inch until the end.

My deepest gratitude goes to my supervisor, Dr. Harith Alani. His deep faith in me, along with his expert guidance and continuous support, were undeniably the main reasons behind all the success and achievements I had out of my PhD work. I am also very thankful to him for teaching me the art of strategic decision making and "bulletproof" academic writing. I could not be more grateful and fortunate to have had Dr. Alani as my PhD supervisor.

My sincere thanks also go to my second supervisor Dr. Miriam Fernandez, who did make herself available for advising and supporting me technically and emotionally during the toughest times of my PhD. The countless brainstorming sessions, during which she patiently spent hours discussing my research ideas and plans, have undoubtedly had a huge impact on the quality of my research.

I am also grateful to my external supervisor Dr. Yulan He, with whom I had the honour to work under direct supervision during the first year of my PhD and her distant supervision afterwards. Dr. Yulan He is one of the most intelligent researchers I have ever worked with. Her extraordinary expertise and techniques in mathematics, machine learning and natural language processing inspired my work on this dissertation.

I would like to extend my gratitude to the members of my dissertation committee, Prof. Markus Strohmaier and Dr. Trevor Collins for their insightful comments and feedback, which helped improve this dissertation. My sincere thanks also goes to the committee chairman, Dr. Paul Mulholland.

Gratitude is also extended to my colleagues in the Knowledge Media Institute (KMi). Their professional support, along with their unique positive spirit and attitude, make KMi one of the best and most unique places to work.

The past few years gave me a chance to meet unique people from all over the globe, who instantly touched my soul with their amazing personality and became an important part of my life. I am extremely grateful to my dearest friends, Giuseppe Scavo, Dra-

homira Herrmannova, Ilaria Tiddi, Lara Piccolo, Lucas Anastasiou, Lukas Zilka, Magdalena Malysz and Maria Maleshkova. Your friendship helped me stay sane throughout these difficult years.

Finally, special recognition goes out to my parents, Riad and Nebal Saif, my brothers Aamer and Rafat, and my sister Sara for supporting me spiritually throughout writing this dissertation and my life in general.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1

INTRODUCTION

The emergence of microblogging services and social media platforms has given web users a venue for expressing and sharing their thoughts and opinions on all kinds of topics and events. Twitter,[1] with nearly 650 million users and over 500 million messages per day,[2] has quickly become a gold mine for organisations to monitor their reputation and brands by extracting and analysing the sentiment of the Tweets posted by the public about them, their markets, and competitors. This new phenomenon has been a motivation for more research efforts, where natural language processing, information retrieval and text processing techniques are utilised to build models and approaches to analyse and track sentiment on Twitter and other similar microblogging services (e.g, Tumbler,[3] Plurk,[4] Twister[5]).

Most existing approaches to sentiment analysis on microblogs proved effective when sentiment is explicitly and unambiguously reflected in the text through affect (i.e. opinionated) words, such as "`great`" as in "`I got my new iPhone 6, such a great device!`" or "`sad`" as in "`So sad, now four Sierra Leonean doctors lost to #Ebola`".

However, merely relying on affect words is often insufficient, and in many cases does not lead to satisfactory sentiment detection results [33]. Common examples of such cases occur when the sentiment of words differs according to (i) the **context** in which those words occur (e.g., "`great`" is negative in the context "`pain`" and positive in the context "`smile`"), or (ii) the **conceptual meaning** associated with the words (e.g., "`Ebola`" is likely to be negative when its associated concept is "`Virus`" and likely to be neutral when its associated concept is "`River`"). Therefore, ignoring the semantics of words when calculating their sentiment, in either case, may lead to inaccuracies, which is a problem that most existing approaches to sentiment analysis on microblogs commonly face.

In this thesis, we investigate the role of words' semantics in sentiment analysis of microblogs, aiming mainly at addressing the above problem. In particular, we research sev-

1 https://twitter.com
2 http://www.statisticbrain.com/twitter-statistics
3 https://www.tumblr.com/
4 http://www.plurk.com/
5 http://twister.net.co/

eral methods for extracting and employing two types of semantics for sentiment analysis: *contextual semantics*, i.e., semantics inferred from co-occurrence of words, and *conceptual semantics*, i.e., semantics extracted from background ontologies and knowledge bases.

We use Twitter as a representative case study of microblogging services in the experimental work conducted in this thesis. Specifically, we study the impact of both types of semantics in multiple sentiment analysis tasks on Twitter. This includes: *entity-level sentiment analysis*, i.e., detecting the sentiment of individual named-entities (e.g., "`Katy Perry`", "`Obama`", etc.), and *tweet-level sentiment analysis*, i.e., detecting the overall sentiment of a given tweet. Also, we investigate the use of semantics for *context-aware sentiment lexicon adaptation*, i.e., amending the prior sentiment orientation of words in general-purpose sentiment lexicons with regards to the words' semantics in the context they occur.

The research work and contribution introduced by this thesis is fivefold:

- We propose a new semantic representation of words, called SentiCircle, that automatically captures the contextual semantics of words in tweets and calculates their sentiment orientation accordingly. We investigate several methods for using the proposed representation in entity- and tweet-level sentiment analysis tasks as well as in adapting general-purpose sentiment lexicons.

- We extract and use the conceptual semantics of words as features for training sentiment classifiers and study the impact of this approach on the sentiment classification performance at tweet level.

- We investigate using the contextual and conceptual semantics of words together for sentiment analysis. In particular, we propose enriching the SentiCircle representation with the conceptual semantics of words and study what impact this has on sentiment analysis at tweet level.

- We propose an approach that finds patterns of words that share similar contextual semantics and sentiment and uses these patterns as features for training sentiment classifiers for tweet- and entity-level sentiment analysis.

- We investigate the impact of removing words which have low discrimination power (aka stopwords) from tweets on sentiment analysis performance.

To our knowledge, the work in this thesis is one of the first works to introduce and investigate the use of the semantics of words in Twitter sentiment analysis. The reason

behind focusing our work and evaluation on tweets data is due to (i) the global popularity of Twitter in comparison to other microblogging platforms, and (ii) the common use of Twitter in the literature on microblogging sentiment analysis.

In the following subsections, we explain the motivation behind our thesis, and detail our research questions, hypotheses and contributions.

1.1 MOTIVATION

The advent and the rapid growth of social media and microblogging services have dramatically changed the amount of user-generated content on Web, with more and more people sharing their thoughts, expressing opinions, and seeking support on such social environments.

Monitoring and analysing sentiment on microblogs, especially Twitter, provides enormous opportunities for both public and private sectors. For private sectors, for example, it has been observed that the reputation of a certain capital stock, product or company is highly affected by the rumours and sentiment published and shared among users on microblogging platforms and social networks [20, 124]. Understanding this observation, companies realised that monitoring and detecting the public's opinion from microblogs leads to building better relationships with their customers, obtaining better understanding of their customers' needs and providing better response to the dramatic changes in their markets.

For public sectors, recent studies [108, 164, 74, 14, 75, 147] suggest a connection between the sentiment expressed on microblogs towards certain public events, and the development and outcome of these events. For example, aggregated Twitter sentiment has been shown to be correlated to political polls [108, 164, 74], and can be also used to understand how people react to crises such as the Westgate Mall Terror Attack in Kenya [147].

The clear and palpable potential that sentiment analysis of microblogs has introduced to either sector, as shown above, has attracted an increasing research interest to this area in recent years. A large number of approaches and tools have been developed for extracting sentiment from Microblogging data, focusing mostly on Twitter data. Each of these approaches comes with certain strengths and weaknesses when applied to tweets, as will be explained in the subsequent sections.

1.1.1 Sentiment Analysis of Twitter: Gaps and Challenges

Sentiment analysis is usually defined as the task of identifying positive and negative opinions, attitudes and emotions towards some subject or the overall polarity of a document [114].

Sentiment analysis is relatively an old research problem [90] with much research work being done and focused mostly on extracting sentiment from *conventional text*, i.e., formal text found on traditional online media platforms, such as personal blogs and news platforms. Existing approaches to extracting sentiment from conventional text can be categorised as *supervised approaches*, which use a wide range of features and labelled data for training sentiment classifiers, and *lexicon-based approaches*, which make use of pre-built lexicons of words weighted with their sentiment orientations (i.e., positive, negative, neutral) to determine the overall sentiment of a given document (see Chapter 2).

Much progress in terms of sentiment classification performance has been achieved by each approach when applied to conventional text of formal language and well known domains, where labelled data is available for training, or when the analysed text is well covered by the used sentiment lexicon [89]. Nevertheless, extracting sentiment from microblogging data, and specifically Twitter data, using these approaches introduces new challenges, due to several characteristics possessed by Twitter data. For example, supervised approaches are domain-dependent and require re-training with the arrival of new data [6]. Obtaining manually labelled tweets from each new domain is labour-intensive. This is because machine learning algorithms often require a large number of labelled instances for training in order to produce classifiers that are able to generalise from training data to unseen instances. Given the great variety of topics and domains that constantly emerge from Twitter, the aforementioned constraints affect the applicability of such approaches.

Lexicon-based approaches do not require training from labelled data. Instead, they rely on sentiment lexicons, such as SentiWordNet [8] and MPQA subjectivity lexicon [176], for sentiment detection. However, traditional lexicon-based methods tend to be ill-suited for Twitter data. Firstly, because sentiment lexicons are composed by a generally static set of words that do not cover the wide variety of new terms, malformed words and colloquial expressions that constantly emerge on Twitter (e.g, "`looov`", "`luv`", "`gr8`"). Secondly, these methods usually make use of the lexical structure of a sentence to determine its sen-

timent, which becomes problematic in Twitter, where ungrammatical structures are very common due to the 140-character length limit [160].

From the above, one can conclude that traditional sentiment analysis approaches need adaptation before they can be efficiently applied to Twitter data and other similar microblogging data sources. This has led to the development of sentiment analysis approaches that take into account the characteristics of Twitter in order to tackle the aforementioned limitations. For example, to overcome the lack of manually annotated training data, supervised methods on Twitter often rely on the distance supervision approach [60] which makes use of automatically generated training data, where emoticons, such as ":-)" and ":(", are typically used to label the tweets as positive or negative. As for lexicon-based sentiment analysis, recent methods are designed [106, 160] to overcome the problem of the language informality, noisy and ill-structured sentences in tweets by relying on lexical rules, such as the existence of emoticons, intensifiers, negation and booster words (e.g., "`Very`", "`Extremely`"), as well as using pre-defined lists of abbreviations and short-forms that are often found in tweets (e.g., "`gr8`", "`luv`").

Overall, most existing sentiment analysis approaches on Twitter adapt fairly well to some of tweets' special characteristics, and therefore produce relatively higher performances than those tuned to work on conventional text. Nevertheless, these approaches are *semantically weak*, that is, they generally do not account for the semantics of words when calculating their sentiment or the sentiment of tweets they occur within [33, 55].

Semantics is generally concerned about exploring what a word or an expression is supposed to mean in a given piece of text [42]. In sentiment analysis, understanding the meaning or the semantics of the word will likely strengthen our understanding of the type of the sentiment or the emotion that this word represents in the text it occurs, as will be explained subsequently.

1.1.2 From Affect Words to Words' Semantics

In the previous section we showed that, existing approaches to Twitter sentiment analysis (machine learning and lexicon-based) usually address some of the challenges and limitations introduced by Twitter data. However, most of these approaches face a common problem, that is: **they are fully dependent on the presence of affect words or syntactical features that explicitly reflect sentiment**. Nonetheless, sentiment in practice is usually conveyed through the latent semantics or meaning of words in texts [139, 24].

For example, The underlying semantics of the word "`great`" in "`I had a great pain in my lower back this morning :(`" does not reflect a positive sentiment. Also, the words "`Ebola`" and "`ISIS`" (Islamic State in Iraq and Syria) in "`Ebola is spreading in Africa and ISIS in Middle East!`" implicitly denote a negative sentiment. However, both supervised and lexicon-based approaches on Twitter are likely to fail in spotting the correct sentiment for these words in such examples, and consequently the overall sentiment of both sentences. This happens mainly because:

- either approach in the first sentence does not account for the semantics of the word "`great`" in the **context** it occurs (e.g., "`pain`"). As a result, the existence of "`great`" is considered positive even though it describes the negative word "`pain`".

  This type of semantics is usually known as *contextual semantics*, since it is typically inferred from the co-occurrence patterns of words in a given context [177, 167].

- either approach in the second sentence, does not consider the explicit semantics of the words "`Ebola`" (i.e., "`Virus/Disease`") and "`ISIS`" (i.e., Islamist Militant Group), which commonly denote a negative sentiment. As a results, the sentence is wrongly assigned a neutral sentiment.

  This type of semantics is known as *conceptual semantics* as it refers to the semantic concepts of words obtained from background knowledge sources such as ontologies and semantic networks [136].

Thus, considering both contextual and conceptual semantics is rather important when detecting the sentiment that words express in tweets. This is especially crucial for words that often change their contextual semantics (e.g., "`great pain`", "`great smile`") or their conceptual ones (e.g., "`Ebola`" -> "`Virus`", "`Ebola`" -> "`River`") as explained above.

The above limitation of traditional or non-semantic approaches has recently brought an increasing interest in the use of semantics in sentiment analysis (aka, *semantic sentiment analysis*). In particular, several approaches that use the contextual or the contextual semantics of words were proposed. These approaches have shown success in capturing the precise sentiment that words indicate in texts, producing therefore better performances than traditional or non-semantic approaches, when applied to conventional text. [166, 97, 87, 24, 55]. Nevertheless, semantic approaches (contextual and conceptual) are generally not tailored to Twitter and the like. First, they are limited by their underlying

lexical or semantic resources (i.e., sentiment lexicons, semantic ontologies), which is especially problematic when processing general Twitter streams, with their rapid semiotic evolution and language deformations [139]. Secondly, these approaches are designed to mainly function on conventional text of formal language and well-structured sentences that Twitter data generally lack, as described earlier.

The work presented in this thesis addresses the problem of building semantic sentiment analysis models on Twitter. Our models are tailored to Twitter data and employ both contextual and conceptual semantics in their workflow, aiming to capture the sentiment of words with regards to their semantics, and consequently improve the overall performance of sentiment analysis.

In the rest of this chapter we state the research questions that we address in this thesis, we describe our research hypotheses, we present our contributions to the state of the art, and we provide the outline of the thesis.

1.2 RESEARCH QUESTIONS, HYPOTHESES AND CONTRIBUTIONS

The main research question investigated in this thesis is:

Could the semantics of words boost sentiment analysis performance on Twitter?

The main focus, as noted, is towards improving the performance of Twitter sentiment classifiers by developing solutions that consider the semantics of words in their sentiment detection workflow. Given the dimensions of the sentiment analysis problem discussed earlier (i.e., the type of the sentiment analysis approach and the type of semantics used), we aspire to achieve improvement in sentiment detection performance by addressing the three following research sub-questions:

RQ1 **Could the *contextual semantics* of words enhance lexicon-based sentiment analysis performance?**

Typical Lexicon-based methods on Twitter assign sentiment to words regardless of their context, i.e., regardless of their contextual semantics (e.g., "`great`" is assigned a positive sentiment in both the contexts "`pain`" and "`smile`"). One assumption to address this limitation is that lexicon-based methods that consider the contextual

semantics of words when calculating their sentiment, perform better than those that do not.

Our hypothesis for extracting and using contextual semantics in sentiment analysis is:

H1 *Context-driven semantics of words can enhance lexicon-based sentiment analysis performance.*

Thus, to test the above hypothesis we propose a dynamic representation of words that captures their semantics from their context and update their sentiment orientations accordingly. The proposed representation first derives the contextual semantics of a word from its co-occurrences with other words, as inspired by the distributional hypothesis-"*Words that occur in similar contexts tend to have similar meanings*" [65, 52, 177]. Next, the contextual semantics of the word is used to calculate its overall sentiment.

We assess the effectiveness of the proposed representation for lexicon-based sentiment analysis on Twitter in three tasks: *entity-level sentiment analysis*, *tweet-level sentiment analysis* and *context-aware sentiment lexicon adaptation*. We design several methods that use the proposed representation under each of these tasks.

Hence, the contribution of our work under this research question can be summarised as follows:

- Propose a semantic representation model, called SentiCircles, that dynamically captures the contextual semantics and sentiment of words from their context.
- Introduce several lexicon-based and rule-based methods, based on SentiCircles, for entity and tweet levels sentiment analysis, and sentiment lexicon adaptation.
- Build and release a new gold-standard dataset for evaluating our proposed methods in each sentiment analysis task.

RQ2 **Could the *conceptual semantics* of words enhance sentiment analysis performance?**

As mentioned earlier, sentiment is often not directly detectable via affect words, but instead associated with the words' conceptual semantics in tweets.

Our proposition for addressing the above question is to detect the sentiment of terms based on their conceptual semantic similarities to other terms whose senti-

ment is known. For example, the entities "iPad", "iPod" and "Mac Book Pro" frequently appear with positive sentiment in a given Twitter corpus and they are all mappable to the semantic concept "Apple Product". As a result, an entity such as "iPhone", whose sentiment might not be known, is more likely to have a positive sentiment since it is also mappable to the concept "Apple Product".

Thus, our hypothesis for this question is the following:

H2 *Semantic concepts tend to correlate with positive or negative sentiment, and hence can be used to determine the sentiment of terms under those concepts.*

Therefore, to address this research question, we propose to extract and incorporate the semantic concepts (e.g. "Person", "Company", "City") of named entities (e.g. "Steve Jobs", "Vodafone", "London") found in tweets into both supervised machine learning and lexicon-based methods to enhance their performances.

For the supervised machine learning approach, we propose adding semantic concepts as features into the training set of sentiment classifiers in order to measure the correlation of the representative concepts with negative/positive sentiment. For the lexicon-based approach, we consider integrating the concepts into our contextual semantic representation, SentiCircles and study the impact on the overall sentiment detection performance.

Our contributions under this line of work are:

- Introduce and investigate the use of conceptual semantics in supervised and lexicon-based sentiment analysis on Twitter.
- Implement three methods for using the conceptual semantics of words as features to train supervised sentiment classifiers.
- Propose a new method, based on SentiCircles, for incorporating the conceptual semantics of words into lexicon-based sentiment analysis on Twitter.

RQ3 **Could *semantic sentiment patterns* boost sentiment analysis performance?**

People tend to convey sentiment in certain contextual semantic structures or patterns rather than in individual words. For example, it is likely that similar word distributions and word co-occurrences (Patterns) are used to express sentiment towards "Beatles" and "Katy Perry", which are likely to differ from the patterns for "iPod" and "McDonald's". Knowing and extracting these patterns from text might help in boosting the performance of sentiment classification methods on Twitter.

Our hypothesis here is:

H3 *Patterns capturing words with similar contextual semantics and sentiment can be used to enhance sentiment analysis performance.*

To address the above question, we propose building a new approach for extracting patterns of words of similar contextual semantics and sentiment from tweets data and using these patterns as classification features for supervised sentiment classifier training. Compared to other works on pattern-based sentiment analysis (see Chapter 5), our approach does not rely on the syntactic structure of tweets, nor requires pre-defined sets of syntactic or semantic templates for pattern extraction. Evaluation on our proposed patterns covers testing their effectiveness in tweet- and entity-level sentiment analysis.

Contributions of this line of work are:

- Propose a novel approach that automatically extracts patterns from the contextual semantic and sentiment similarities of words in tweets.
- Use patterns as features in tweet- and entity-level sentiment classification tasks.
- Conduct quantitative and qualitative analysis on a sample of our extracted semantic sentiment patterns and show the potential of our approach for finding patterns of entities of controversial sentiment in tweets.

RQ4 **What effect does stopword removal have on sentiment classification performance?**

Words are usually characterised by their discrimination power, which often determines their contribution towards making correct classification decision. In sentiment analysis, words of low discrimination power in a certain context are considered stopwords and get removed from the analysis accordingly.

Several stopword extraction and removal methods have been used for sentiment analysis of conventional text [163, 7, 148]. However, the impact of these methods on Twitter sentiment analysis is not well explored.

Our hypothesis for this question is:

H4 *Different stopword removal methods could have different impact on sentiment analysis performance on Twitter.*

Thus, to address the above question, and as an additional contribution of this thesis, we analyse the effectiveness of several stopword removal methods for senti-

ment classification of tweets and whether removing stopwords influences the performance of Twitter sentiment classifiers. To this end, we apply six different stopword removal methods to Twitter data from six different datasets and observe the impact of removing stopwords on supervised sentiment classification methods. We assess such impact by observing fluctuations in: (i) the level of data sparsity, (ii) the size of the classifier's feature space, and (iii) the classifier's performance in terms of accuracy and F-measure.

Our contribution under this line of work is summarised as follows:

- Conduct an analysis study on the impact of several stopword removal methods in Twitter Sentiment analysis using different Twitter datasets and sentiment classifiers.
- Measure fluctuations of removing stopwords on the level of data sparsity, the size of the classifier's feature space and its classification performance.
- Show that, despite the popular use of pre-compiled stoplists in Twitter sentiment analysis, they can have negative impact on the sentiment classification performance.
- Show that, in practical applications for Twitter sentiment analysis, the ideal stopword generation method is the one that removes those infrequent terms appearing only once in the Twitter corpus.

1.3 THESIS METHODOLOGY AND OUTLINE

The main purpose behind our work in this thesis is to improve the performance of Twitter sentiment analysis by using both, the contextual and conceptual semantics of words. To this end, we design a general methodology that we use in the different parts of this thesis. Our methodology consists of three main steps; extraction, incorporation and assessment, as depicted in Figure 1 and detailed below:

1. Extraction: design methods for capturing the contextual and conceptual semantics of words from a given collection of tweets.
2. Incorporation: investigate several methods and models for incorporating and using both types of semantics in sentiment analysis.

3. Assessment: measure the performance of our semantic methods in multiple sentiment analysis tasks on Twitter; entity-level sentiment analysis, tweet-level sentiment analysis, and sentiment lexicon adaptation.

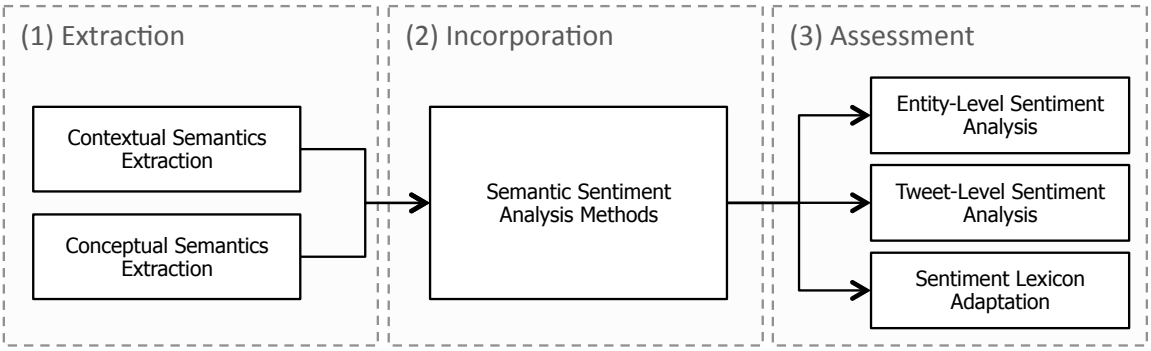


Figure 1: General methodology for extracting, incorporating and assessing the effectiveness of words' semantics in Twitter sentiment analysis.

Note that we experiment with our proposed methods in each of the three steps using different settings, i.e., several Twitter datasets and different sentiment lexicons. As for evaluation and assessment, we compare against several state-of-the-art Twitter sentiment analysis methods.

Figure 2 shows the general pipeline of our work with regard to the core contribution chapters of this thesis. As can be noted, the three-step methodology described above is used in each of these chapters.

The material of this thesis is distributed in individual chapters as follows:

Part I: Background and Literature Review

In **Chapter 2** we provide a fundamental background knowledge of the sentiment analysis task, subtasks, levels and popular approaches. After that, we present a comprehensive overview of the existing work in the the area of sentiment analysis on Twitter. Throughout our review, we reveal the strengths and weaknesses of the reviewed work and map them to our research questions and hypotheses.

Part II: Semantic Sentiment Analysis of Twitter

In **Chapter 3** we describe our work on using the contextual semantics of words for improving the performance of lexicon-based sentiment analysis methods on Twitter. We also explore the use of contextual semantics for context-aware adaptation of general-purpose sentiment lexicons. The work in this chapter addresses our first research question.

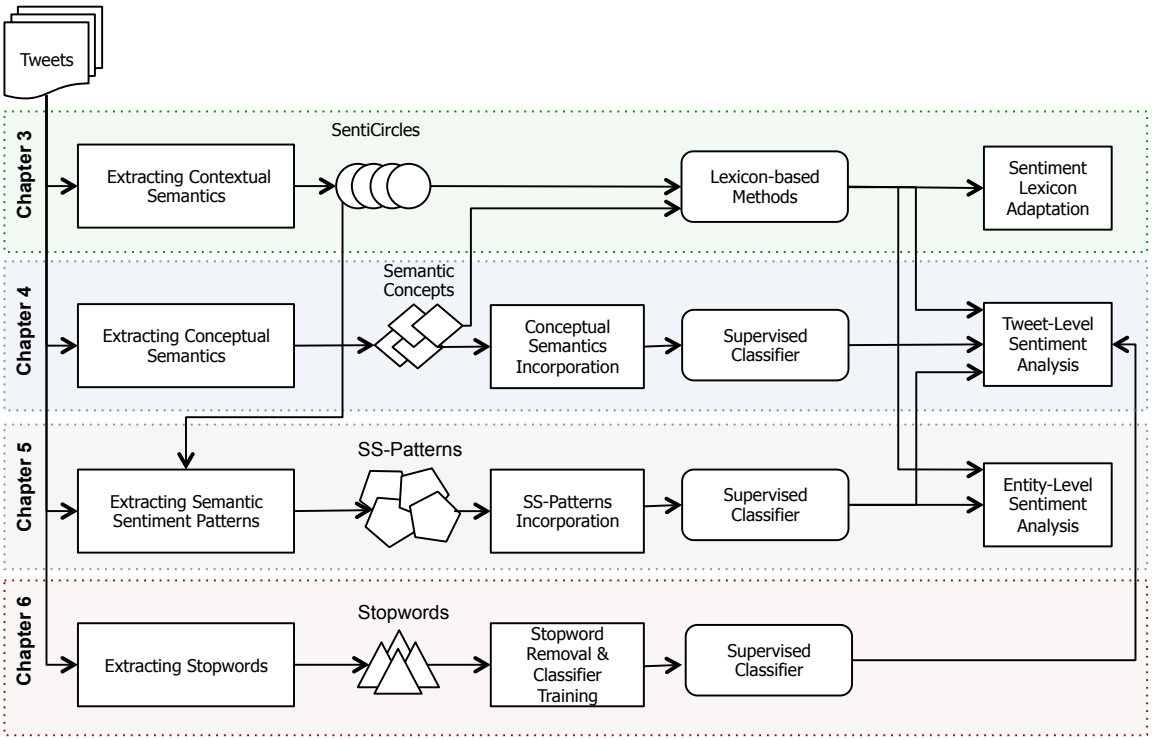


Figure 2: Pipeline of our work under each chapter, and thesis contribution outline. Arrows crossing two different chapters mean that a component developed in one chapter is used in the other one.

In **Chapter** 4 we present our work on using the conceptual semantics of words to improve the performance of both lexicon-based and supervised sentiment analysis approaches on Twitter, addressing thereby, the second research question of this thesis.

In **Chapter** 5 we present our approach for extracting semantic sentiment patterns of words and using these patterns for boosting supervised sentiment classification performance on Twitter. In this chapter we address our third research question.

Part III: Analysis Studies

In **Chapter** 6 we present our analysis on the impact of removing stopwords on the performance of Twitter sentiment classifiers. Our fourth research question is addressed in this chapter.

Part IV: Discussion and Conclusion

In **Chapter** 7 we discuss the work presented in this thesis, highlight our contributions and point out future work.

In **Chapter** 8 we present our main conclusions.

1.4 PUBLICATIONS

Chapters of this thesis are based on the following publications:

- **Chapter 3**
  - Hassan Saif, Yulan He, Miriam Fernandez and Harith Alani (2015). *Contextual Semantics for Sentiment Analysis of Twitter*. Information Processing and Management Journal (IPM).
  - Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani (2014). *SentiCircles for Contextual and Conceptual Semantic Sentiment Analysis of Twitter*. In Proceeding of the 11th Extended Semantic Web Conference (ESWC), Crete, Greece.
  - Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani (2014). *Adapting Sentiment Lexicons using Contextual Semantics for Sentiment Analysis of Twitter*. In Proceeding of the 1st Semantic Sentiment Analysis workshop (SSA2014), Crete, Greece. (*Best Paper Award*)
  - Hassan Saif, Miriam Fernandez, Yulan He and Harith Alani (2013). *Evaluation Datasets for Twitter Sentiment Analysis: A survey and a new dataset, the STS-Gold*. In Proceeding of the 1st Workshop on Emotion and Sentiment in Social and Expressive Media: approaches and perspectives from AI (ESSEM) at AI*IA Conference, Turin, Italy. (*Best Paper Award Nominee*)
- **Chapter 4**
  - Hassan Saif, Yulan He, and Harith Alani (2012). *Semantic Sentiment Analysis of Twitter*. In Proceeding of the 11th International Semantic Web Conference (ISWC), Boston, US.
  - Hassan Saif, Yulan He, and Harith Alani (2012). *Alleviating Data Sparsity for Twitter Sentiment Analysis*. In Proceeding of the 2nd Workshop on Making Sense of Microposts (#MSM2012): Big things come in small packages: in conjunction with WWW, Lyon, France. (*Best Paper Award*)
  - Hassan Saif, Yulan He, and Harith Alani (2011). *Semantic Smoothing for Twitter Sentiment Analysis*. Poster at the 10th International Semantic Web Conference (ISWC), Bonn, Germany. (*Best Poster Award Nominee*)
- **Chapter 5**

  - Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani (2014). *Semantic Patterns for Sentiment Analysis of Twitter*. In Proceeding of the 13 International Semantic Web Conference (ISWC), Trentino, Italy.

- **Chapter 6**

  - Hassan Saif, Miriam Fernandez, Yulan He and Harith Alani (2014). *On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter*. In Proceeding of The 9th International Conference on Language Resources and Evaluation (LREC), Reykjavik, Iceland.
  - Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani (2014). *Automatic Stopword Generation using Contextual Semantics for Sentiment Analysis of Twitter*. Poster at the 13 International Semantic Web Conference (ISWC), Trentino, Italy.

Part I

BACKGROUND

The good opinion of mankind, like the lever of Archimedes, with the given fulcrum, moves the world.

Thomas Jefferson

2

LITERATURE REVIEW

Sentiment analysis has established itself in the past few years as a solid research topic, providing organisations and businesses with solutions to monitoring and analysing the public's opinion towards their products, brands and services. To this end, a vast amount of research has been done, mostly focusing on sentiment detection of *conventional* text such as review data, online blogs and discussion forums. However, the explosion of social networks and microblogging services, has dramatically shifted the research interests towards the analysis of sentiment of microblogging data, and particularly tweets data. Twitter,[1] has become one of the most popular microblogging services, producing content reflecting opinions about topics, products and life-events [76, 110, 47].

Broadly speaking, existing approaches to sentiment analysis can be categorised into two main categories: *traditional approaches* and *semantic approaches*. Traditional approaches rely on the presence of words or syntactical features that explicitly reflect sentiment. Semantic approaches, on the other hand, exploit the latent semantics of words in text to capture their context and update their sentiment orientations accordingly.

In this chapter, we provide an overview of the sentiment analysis literature, targeting mainly those works on Twitter, since analysing sentiment from tweets is the main focus of this thesis. In particular, the main elements and dimensions of the sentiment analysis problem are introduced in Section 2.1. A literature review of the sentiment analysis research on Twitter is provided in Section 2.2. Key existing works on semantic sentiment analysis are reviewed in Section 2.3. Discussion on the main strengths, limitations and gaps in the state-of-the-art of Twitter sentiment analysis is provided in Section 2.4.

2.1 BACKGROUND

In this section we provide the reader with the fundamental background knowledge about sentiment analysis, which constitutes the basis of the research presented in this thesis.

1 https://twitter.com

2.1.1 Fundamentals

Sentiment analysis is the task of identifying positive and negative opinions, emotions and evaluations in text. It is a multidisciplinary task, which exploits techniques from computational linguistics, machine learning, and natural language processing, to perform various detection tasks at different text-granularity levels.

Sentiment analysis is a very rich research problem, with a large amount of work being published every year targeting the various aspects and dimensions of the problem. Broadly speaking, the problem of sentiment analysis can be divided into four main dimensions, as depicted in Figure 3 and summarised below.

1. The sentiment analysis task which varies from polarity detection to more fine-grained tasks, such as emotion detection and sentiment strength detection, as will be explained in the next subsection.

2. The sentiment analysis level which is determined based on the granularity of text used for the analysis (e.g., document-level, sentence-level, phrase-level, etc).

3. The sentiment analysis approach which covers the type of sentiment analysis approach used (e.g., supervised approaches, lexicon-based approaches, hybrid approaches).

4. The data type: sentiment can be extracted from conventional text (e.g., news articles, product reviews) as well as microblogging data (tweet messages, Facebook status updates, SMS, etc).

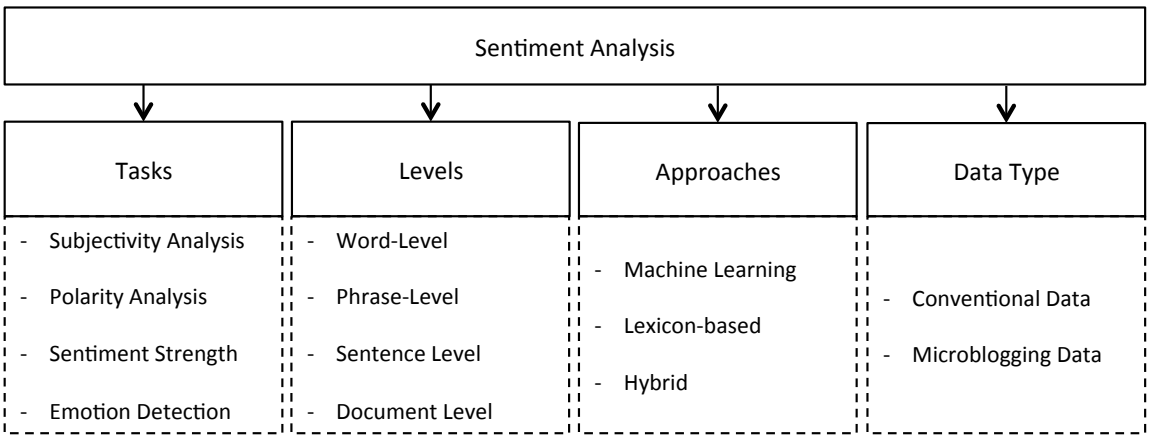


Figure 3: Four dimensions of the sentiment analysis research problem.

The rest of this section describes each of these dimensions in depth, providing references to external resources when necessary. For clarity, we are going to use the following review about the *iPhone 5* as an example[2] to illustrate the different dimensions of sentiment analysis (each sentence in the example is coupled with a number for easy reference):

> *(1) I have recently upgraded to iPhone 5. (2) I am not happy with the screen size. (3) It is just too small :(. (4) My best friend got Galaxy Note 3. (5) He got a much larger screen than mine!!!. (6) Even if his hardware outperforms mine, I easily outrun him with all the apps I can get :)*

1.Sentiment Analysis Subtasks

The basic subtask in sentiment analysis is *polarity detection*, that is, decide whether the sentiment of a given text is *positive* or *negative*. In our example, sentences (2) and (3) have negative sentiment while sentence (6) has a positive sentiment. [3]

Another popular subtask is *subjectivity detection*, which aims to identify whether the text is *subjective* (i.e., has a positive or negative sentiment) or *objective*, (i.e., has a neutral sentiment). For example, sentences (1) and (4) are objective sentences, i.e., they are factual or neutral. On the other hand, sentences (2), (3), (5) and (6) are all subjective.

Subjectivity and polarity detection are, to a large extent, the most popular subtasks in sentiment analysis [89], but not the only ones. Other relevant subtasks include: (i) *emotion detection*, which aims to identify the human emotions and feelings expressed in text, such as "*happiness*", "*sadness*", "*anger*", etc. In our example, sentence (3) shows "*sadness*" emotions while sentence (6) shows "*happiness*" emotions. (ii) *Sentiment strength detection*, which aims at measuring the strength or the intensity of the sentiment in text. For example, this task tries to detect how strong the positive sentiment is in sentence (6) and how strong the negative sentiment is in sentence (3).

2. Sentiment Analysis Levels

Broadly speaking, the aforementioned tasks have been extensively researched in the literature, aiming at analysing the sentiment at four different text granularity levels. Each one of these levels differs from the others in the level of granularity of the analysed text, as follows:

2 The review is obtained from http://reviews.cnet.com/. The reviewer's name is anonymized for privacy purposes.

3 Polarity detection in some work also involve the detection of a third sentiment class, *neutral sentiment.*

- *Word-level* sentiment analysis: given a word w in a sentence s, decide whether this word is opinionated (i.e., express sentiment) or not. If so, detect what sentiment the word has. Note that when the word is considered a named-entity (i.e., the word represents an instance of a predefined category, such as Person, Organisation, Place, Product, etc.), the task is called entity-level sentiment detection. In our example, the entity "`iPhone`" receives a negative sentiment on average while the entity "`Galaxy Note 3`" receives a positive one.

- *Phrase-level* sentiment analysis: also known as *expression-level* sentiment analysis. Given a multi-word expression e in a sentence s, the task is to detect the sentiment orientation of e. For example, the expression "`I am not happy`" in sentence (2) has a negative sentiment.

- *Sentence-level* sentiment analysis: given a sentence s of multiple words and phrases, decide on the sentiment orientation of s. For example, sentence (2) has a negative sentiment while sentence (6) has a positive sentiment.

- *Document-level* sentiment analysis: given a document d, decide on the overall sentiment of d. This is usually done by averaging the sentiment orientation of all sentences in d.

It is worth noting that when a word or a phrase describes a specific aspect or feature of a common sense object (e.g., product) in text, the sentiment analysis task is called *aspect-level sentiment analysis*. Aspect-level sentiment analysis is defined as: given a document d, an object o and a set of of aspects A, detect the sentiment expressed in d towards each aspect $a_i \in A$ of the object o. For example, in sentence (2) the aspect "`screen size`" of "`iPhone`" has a negative sentiment while the same aspect of "`Galaxy Note 3`" has a positive sentiment.

3. Sentiment Analysis Approaches

Early research efforts in sentiment analysis go back to the late-seventies with the work of Jaime Carbonell in belief models and systems [35]. However, the major take-off of the research was in the early 2000 along with the explosion of Web 2.0, which helped increase the awareness of the importance of the field and its potentials for business, economics, politics and online marketing areas. Extensive research efforts in sentiment analysis have been conducted during the last two decades resulting in a large amount of models and

methods being proposed. These methods have generally followed two main approaches, the *machine learning* and *lexicon-based* approach:

- *The machine learning approach*: this approach considers the problem as a typical text classification problem. A machine learning classifier, such as Naive Bayes (NB) or Maximum Entropy (MaxEnt) [18], is trained from a corpus of pre-annotated data (i.e., a collection of textual elements, such as documents or sentences, annotated with their sentiment labels). Once trained, the classifier can be applied to infer the sentiment of unseen documents (i.e., documents of unknown sentiment labels). Pioneering work in this vein was published in 2002 by Pang and Lee [115] performing polarity classification (positive vs. negative) on a movie review dataset.[4] To this end, the authors used three supervised classifiers, SVM [66], Naive Bayes and Maximum Entropy (MaxEnt). Many variations to this approach have since been proposed [113, 44, 173, 125] following the same training/inferencing principle.

- *The lexicon-based approach*: this approach, on the other hand, assumes that the sentiment orientation of a given document is the average of the sentiment orientations of its words and phrases [165]. This approach relies on sentiment lexicons (i.e., prebuilt dictionaries of words with associated sentiment labels) in order to extract the opinionated words in the documents and use them to calculate its overall sentiment accordingly. To this end, several lexicons have been built such as SentiWordNet [8], MPQA subjectivity lexicon [176] and the LIWC lexicon [118].

4. Sentiment Analysis Data Types

Along the last two decades of sentiment analysis, the proposed sentiment analysis methods have been applied to several types of textual data coming from different web sources such as review websites, online blogs, news portals, discussion forums, social networks (Facebook,[5] MySapce,[6] etc.), microblogging services (e.g., Twitter, Tumbler,[7] etc.), etc. Most of the data published by these sources can be divided into two main categories with respect to their characteristics. These categories are (i) *conventional data* and (ii) *microblogging data*:

- *The Conventional Type of Data*:

4 http://www.cs.cornell.edu/people/pabo/movie-review-data/
5 https://www.facebook.com/
6 https://myspace.com/
7 https://www.tumblr.com/

Early work on sentiment analysis focused on detecting the sentiment of text found in review data (e.g., product reviews), online blogs, discussion forums and news articles. Several resources were built to this end. This includes the Cornell movie review dataset [113], the customer review dataset [71], the MPQA corpus [174], and the NTCIR multilingual corpus [143]. Two main characteristics are commonly attributed to these resources. First, they are all composed of formal, well-written and well-structured documents. Secondly, they are all manually annotated for sentiment by human coders. We refer to these type of data as *conventional data* or *conventional text*. We also refer to the sentiment analysis on this type of data as *sentiment analysis over conventional text* or *conventional sentiment analysis*.

- *The Microblogging Type of Data*:

  The evolution of microblogging services and social networks in the last decades,[8] such as Twitter and Facebook, has created a new phenomenon. The amount of the user-generated content on the web has increased exponentially, with literally millions of tweets and status updates being published every hour. Such extraordinary phenomenon has a big impact on the sentiment analysis research community, where the interest in sentiment analysis has drifted away from analysing conventional data to the analysis of microblogging data. This new research direction is known as *Sentiment Analysis of Microblogs*.

  Unlike conventional text, microblogging text is usually very short (e.g., tweet messages are limited to 140 characters), lack of complete and correct sentence structures, and contains many abbreviations, ill-formed words and irregular phrases.

  Despite the wide spread of microblogging services on the web, most of the existing work on microblogging sentiment analysis focuses on the sentiment analysis of tweets data. This research area is known as *Sentiment Analysis of Twitter* [60, 11, 16, 82, 110, 1, 123, 45, 22, 5, 61, 116, 10, 4]. In addition to the works focused on analysing Twitter data, a number of studies exist on sentiment analysis over other microblogging services and social media platforms such as MySpace [159, 111], Facebook [83, 84], Sina Weibo [69],[9] etc.

8 The real take-off of microblogging services was by the official launch of the Facebook *Status Updates* service in March 2005.

9 http://weibo.com/

2.1.2 A Note on Terminology

This chapter introduces a large number of research terms, phrases and acronyms that are often used in sentiment analysis. Some of these terms might be known to the reader while others might not. This document tries to cover each of these terms with an inline, footnote, or side-note description according to their importance and complexity.

The reader might notice that some of these terms are interchangeably used in different contexts, yet they refer to the same thing. Below are the most popular cases:

- Sentiment *analysis*, sentiment *classification* and sentiment *detection* refer in this document to the same task. However, *sentiment analysis* usually refers to the main research area while *sentiment classification* and *sentiment detection* are often used in supervised machine learning and lexicon-based approaches respectively.

- Sentiment *class*, sentiment *label*, sentiment *orientation* and sentiment *score* refer to the sentiment value of a given text. While sentiment *class*, *label* and *orientation* are literal values (e.g., positive, negative, neutral), sentiment *score* is usually a numerical value (e.g., -1 denotes negative sentiment and +1 denotes positive sentiment).

In the rest of this chapter, we provide a thorough review of existing work in the Twitter sentiment analysis literature, followed by a deep discussion on the advantages and limitations of these works. The identified gaps and limitations motivate the research work presented in this thesis.

2.2 SENTIMENT ANALYSIS OF TWITTER

Twitter is an online microblogging service created in March 2006. It enables users to send and read text-based posts, known as tweets, with 140-character limit for compatibility with SMS messaging. As of July 2014, Twitter has about 270 million active users generating 500 million tweets per day.[10]

Tweet messages usually convey sentiment [110]. However, unlike conventional data, tweets have certain characteristics, that make the problem of analysing sentiment over Twitter data a harder problem than the one of analysing conventional data. Tweet messages are too short encouraging the use of abbreviations, irregular expressions, poor grammar and misspellings. Such characteristics make it hard to identify the opinionated content in tweets using sentiment analysis approaches designed for conventional text [137].

Aiming to target the above problem, research work has been conducted, especially in the past four years, focusing on the particular problem of sentiment analysis over Twitter data. Similar to conventional sentiment analysis, existing approaches to Twitter sentiment analysis can be divided into machine learning and lexicon-based approaches.

In the second part of this chapter we describe existing works on Twitter sentiment analysis under the machine learning approach (Section 2.2.1.1) and lexicon-based approach (Section 2.2.1.2) separately, exploring their strengths and weaknesses. Moreover, since the main goal of this thesis is towards semantic sentiment analysis, we will further divide the literature into *traditional approaches*, i.e, those ones that do not consider semantics (Section 2.2.1), and *semantic approaches*, i.e., those ones using semantics (Section 2.3). Our literature review of semantic sentiment analysis approaches in Section 2.3 covers not only Twitter specific works, but also approaches focused on conventional text. This is because (i) the underlying concept of using semantics is pretty much the same for both tasks, and (ii) the lack of research work on semantic sentiment analysis on Twitter.

10 https://investor.twitterinc.com/releasedetail.cfm?ReleaseID=862505

2.2.1 Traditional Sentiment Analysis Approaches

2.2.1.1 *The Machine Learning Approach*

The machine learning approach to sentiment analysis is characterised by the use of a machine learning algorithm and a training corpus. The approach usually functions in two phases, a *training phase* and an *inference phase*, as illustrated in Figure 4.[11] In the training phase a machine learning algorithm is used to train the classification model using a set of features extracted from the tweets of the training corpus. In the inference phase the trained model is used to infer the sentiment label of unseen tweets (i.e., tweets of unknown sentiment classes).

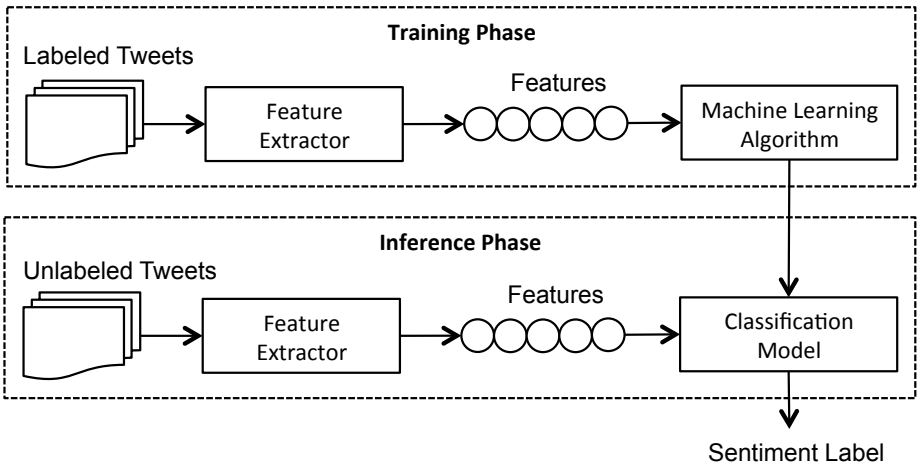


Figure 4: Pipeline of the machine learning approach to sentiment analysis on Twitter.

Machine learning models are often divided into three main categories based on the supervision required during the training (learning) phase:

1. *Supervised classifiers* require training from *labelled* samples (e.g., tweets labelled with their sentiment class). Naïve Bayes (NB), Maximum Entropy (MaxEnt) and Support Vector Machine (SVM) are well-known examples under this category [81].

2. *Unsupervised classifiers* work with *unlabelled* samples. Popular unsupervised algorithms include clustering algorithms (e.g., k-means, k-medoids, hierarchical clustering), hidden Markov models and some unsupervised neural network models such as self-organising maps [18].

11 Creation of this figure is inspired by a previous figure from the NLTK toolkit (http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html).

3. *Semi-supervised classifiers* require both labelled and unlabelled samples. Algorithms like label propagation and graph-based models are in this category [187].

The performance of the machine learning approach is generally affected by three main factors, as shown in Figure 5. These factors are: (i) the training methodology, (ii) the type of features used to describe/characterise the data, and (iii) the choice of the machine learning classifier (algorithm). Therefore, we review research work in this section with respect to these three factors.

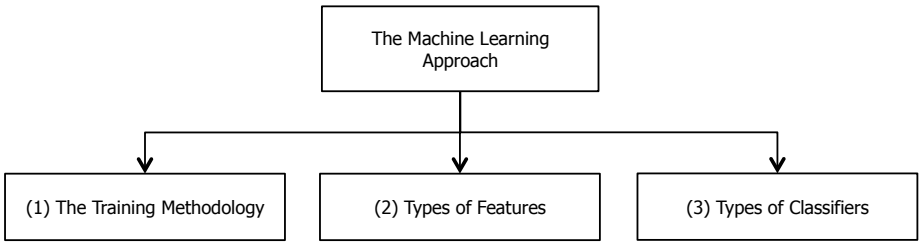


Figure 5: Three main factors that affect the sentiment classification performance of the machine learning approach.

1. The Training Methodology

Most exiting works in Twitter sentiment analysis under the machine learning approach use supervised classifiers, and therefore, they require labelled corpora for classifier training. In Section 2.1.1 we saw that previous works on conventional sentiment analysis [115, 113, 97, 173, 183] tend to train their models from manually labelled corpora, such as the movie review dataset,[12] which was constructed by Pang and Lee [115, 113] and the Customer Review dataset,[13] introduced by Hu and Liu [71].

Similar to conventional text, generating manually labelled Twitter data is a labour-intensive task. In addition, the use of abbreviations, colloquial terms and ill-formed words may make the annotation of Twitter data even more difficult than the annotation of conventional text [123]. Aiming to overcome this problem, several studies on Twitter sentiment analysis [60, 11, 16, 110, 45, 82, 123, 156] have relied on the *distant supervision* [125] approach to automatically construct large corpora of annotated tweets for sentiment classifier training. The distant supervision approach assumes that emoticons,[14]

12 http://www.cs.cornell.edu/people/pabo/movie-review-data/
13 www.cs.uic.edu/~liub/FBS/Reviews-9-products.rar
14 http://en.wikipedia.org/wiki/Emoticon

hashtags,[15] emoji,[16] etc., are good indicators to the sentiment in tweets and can be used to automatically assign sentiment labels.

Go et al. [60], for example, used emoticons such as ":), :-), :D" and ":(, :-(, : (" to construct a training corpus of 1.6 million positive and negative tweets. The corpus was made public and used by the authors to train supervised classifiers for polarity sentiment classification.

Pak and Paroubek [110] generated a corpus of tweets of three sentiment classes (positive, negative and neutral). As in [60], positive and negative tweets were annotated using emoticons, while neutral tweets were obtained by crawling several newspapers accounts on Twitter (e.g., New York Times and Washington Posts).

In addition to emoticons, other works proposed using hashtags as additional indicators of tweets' sentiment [45, 82]. Kouloumpis et al. [82] utilised three sets of positive, negative and neutral hashtags (e.g., *#bestfeeling*, *#worst*, *#news*) to label tweets extracted from the Edinburgh Twitter corpus,[17] where each tweet is annotated based on the polarity of these hashtags. Evaluation results showed that an AdaBoost.MH classifier [141], trained from tweets labelled using hashtags solely, gives lower performance than the one trained from tweets labelled using a combination of hashtags and emoticons. In a similar vein, Davidov et al [45] used 50 different hashtags such as "*#sucks*", "*#obama*", "*#sarcasm*" as direct labels to tweets extracted from the O'Connor Twitter corpus [108]. Each hashtag was used to annotate 1000 tweets resulting in a training dataset of 50K tweets of 50 different sentiment classes (each class corresponds to a single hashtag representing a particular sentiment emotion). The annotated corpus was used to perform a 50-class classification task, as well as a binary-hashtag classification task (i.e., check whether the tweet expresses any of the fifty selected emotions or not).

In addition to polarity and subjectivity classifications, the distant supervision approach was used to build training corpora for *emotion detection* on Twitter [123, 156]. Emotion detection is usually a multiple classification task (see Section 2.1.1) and requires training corpora containing tweets of multiple emotion classes (e.g., *happy, sad, anger, fear, joy*). Purver and Battersby [123] exploited the use of emotions and hashtags as noisy labels to annotate tweets with the six basic emotion classes defined by Ekman [51] (*happy, sad, anger, fear, surprise, disgust*). To this end, the authors used a collection of tweet messages, in which each tweet contains, at least, an emoticon or a hashtag that corresponds to one

15 http://en.wikipedia.org/wiki/Hashtag
16 http://en.wikipedia.org/wiki/Emoji
17 http://demeter.inf.ed.ac.uk

of the six emotion classes. For example, the emotion class "*happy*" was assigned to tweets containing either the hashtags *#happy* or *#happiness*, or an emoticon such as ":-)" or ";-)". The authors performed a cross-validation test on the produced labels. In particular, several classifiers were trained from tweets labelled with emoticons, and tested on tweets labelled with hashtags and vice versa. The goal was to check whether hashtags and emoticons assigned to similar emotion class are good indicators to the actual emotional state conveyed in tweets. Evaluation results came back with a positive answer; both, emoticons and hashtags are indeed good indicators of the emotional states in tweets. However, the results also showed that the classification performance was higher (60-65% range) for the emotion classes *happiness*, *sadness* and *anger* than for the classes *fear*, *surprise* and *disgust*. The authors argued that the low performance of the latter classes was because these classes are generally negative and they often share similar emoticon and hasghtag indicators.

Recently, Suttles and Ide [156] have exploited the use of *emoji*[18] as noisy labels in addition to emoticons and hashtags. Similar to [123], the task was to build an annotated Twitter corpus for emotions classification. However, the authors proposed classifying emotions into eight different, yet bipolar classes identified by Plutchik [120] (*joy* vs. *sadness*, *anger* vs. *fear*, *trust* vs. *disgust*, and *surprise* vs. *anticipation*). For data annotation, the authors used a combination of 69 emoticons, 70 emoji and 56 hashtags. A cross-validation test on these labels showed similar results to [123]: (i) using all the three types of labels for annotation resulted in reliable performance, and (ii) the performance tended to be higher for *joy* and *sadness* classes than for other classes.

Barbosa and Feng [11] followed a slightly different approach. Instead of using hashtags or emoticons as noisy labels, they collected and annotated their training data using three third-party Twitter sentiment detection services including Twendz,[19] TweetFeel,[20] and Sentiment140.[21] While Twends and TweetFeel use pre-built sentiment lexicons to label each tweet as positive or negative, Sentiment140 uses a MaxEnt classifier trained from the same Twitter corpus used in [60]. The annotation agreement between each pair of sentiment detection services was between 40% and 60%. The authors also found that the best strategy to enhance the labelling results is by combining the labels from the three

18 emoji are smileys which were used first in Japanese oneline media and text messages in 1998.

19 http://twendz.waggeneredstrom.com

20 http://www.tweetfeel.com

21 http://www.sentiment140.com/

websites. Evaluation results showed that such a strategy gives a higher classification performance than using each service solely.

From the above review, it can observed that the distant supervision approach was, to a large extent, successfully used for the automatic sentiment annotation of tweets. Classifiers trained from automatically labelled corpora achieved performances as high as 83% in accuracy [60]. However, the distant supervision approach is based on the assumption that the sentiment of a tweet corresponds to the sentiment of the hashtags or emotions within it. This assumption is susceptible to errors when dealing with tweets that either (i) do not contain any sentiment indicators, (ii) contain multiple, yet contradictory, sentiment indicators, or (iii) when an emoticon has a different sentiment to the tweet it belongs.

Due to the above limitations, several works argued that the distant supervision approach is often inaccurate and may harm the performance of sentiment classifiers [152, 91]. Instead, they proposed using approaches which, either do not use noisy sentiment labels (i.e., emoticons, hashtags, etc.) at all, or lower the dependency on them when training sentiment classifiers.

Liu et al. [91] trained their classifiers from a combination of noisy and manually labelled tweets. In particular, the authors first trained a Bayesian classifier from a small set of manually labelled tweets. After that, they interpolated the language model of the classifier with an another model trained from noisy labels (emotions) as follows:

$$P_{co}(W|C) = \alpha P_a(W|C) + (1-\alpha)P_u(W|C) \quad (1)$$

Where $P_{co}(W|C)$ is the interpolated language model, $P_a(W|C)$ is the language model trained from manually labelled tweets, and $P_u(W|C)$ is the language model trained from noisy labelled tweets. The authors trained and evaluated their model using the Sanders Twitter corpus [138]. Evaluation results showed that the smoothed model outperforms consistently other baseline models not considering interpolation.

Speriosu et al. [152] decided not to use noisy labels. Instead, they proposed a label propagation method on Twitter follower graphs. First, a graph was constructed with users, tweets, word unigrams, word bigrams, hashtags, and emoticons as nodes, which are connected based on the link existence among them (e.g., users are connected to the tweets they created; tweets were connected to the word unigrams they contained, etc.). A label propagation method was then applied, where sentiment labels were propagated from a small set of nodes, manually assigned a sentiment label, throughout the graph. They claimed that their label propagation method outperforms MaxEnt trained from noisy labels.

2. Types of Features

The second factor, which affects the performance of sentiment classifiers, is the choice of the features used for classifier training. Many types of features have been used in Twitter sentiment analysis, including (i) word n-gram features,[22] (ii) lexicon features, (iii) Part-Of-Speech tags (POS) features, and (iv) microblogging features. [60, 116, 110, 11, 13, 45, 1, 82, 123].

Go et al. [60] used word unigrams and bigrams in conjunction with POS tags in the training of NB, MaxEnt and SVM classifiers. They found that a MaxEnt classifier trained from a combination of unigrams and bigrams outperforms other models trained from other combinations of features by nearly 3%. However, a contrary finding was reported by Bermingham and Smeaton [13]. They found that the best performance, based on a different dataset to the one used in [60], was achieved by a NB classifier trained from word unigrams only.

In a similar vein, Pak and Paroubek [110] investigated the POS tag features in addition to three different variations of word n-grams, unigrams, bigrams and trigrams. However, They did not include the POS tag features into the feature space directly. Instead, they used the distribution of these features within the tweet messages of a given sentiment class (i.e., $P(POS|C)$) in order to calculate the posterior probability of a NB classifier trained from word n-grams. Evaluation results showed that, using bigram features for training gives the highest performance compared to using unigrams or trigrams. The authors also argued that their NB model, which is augmented by the POS distributions in tweets, outperforms SVM and CRF classifiers trained from word n-grams only.

Despite the popular use of word n-grams in the previous work, these features have been argued to hinder the classification performance because of the large number of infrequent words in Twitter, which make the data very sparse [4, 11, 5]. Instead, features like *character n-grams* and microblogging features have been investigated. Aisopos et al. [4] compared between word n-gram and character (letter) n-gram features. They argued that the latter features have an advantage over the former. Specifically, they found that character n-grams are more tolerant to noise than word n-gram. The authors also proposed using a n-gram graph representation [58] of tweets rather than the simple vector representation. Nodes in this graph correspond to the character n-grams in a given tweet,

22 In the tweet "`I like my iPhone! :)`", any single word represents *word unigram* (e.g, `iPhone`), any 2 adjacent words represent a *word bigram* (e.g., `my iPhone`), any 3 adjacent words represent a *word trigram* (e.g., `like my iPhone`), and so on.

while the edges represent the adjacency information (distance) between all pairs of nodes. (i.e., the distance between two n-grams in a given tweet). The n-gram graph representations as well as the ordinary vector representation of word n-grams were evaluated using NB and C4.5 classifiers on a collection of 475 million tweets from Yang and Leskovic [181]. Evaluation results, however, were not as conclusive; using n-gram graphs with C4.5 classifier gives a higher performance than the vector model, but a lower performance when the NB classifier is used.

Asiaee et al. [5] claimed that a high classification performance of tweets' sentiment is still achievable by using the word n-gram features. Their trick, however, was to covert the high dimensional back-of-word vector representation ($\mathbf{v}$) into a sparse vector representation ($\mathbf{x}$) of much lower dimensionality. To this end, they used a random reconstructible projection (RP) technique [17], where tweets are represented as a random matrix (P) in a way that the sparse vector representation of a given tweet (t) is computed as: $\mathbf{x_t} = \mathbf{P v_t}$. Evaluation results showed that training supervised classifiers such as SVM, NB and KNN from the new vector representation leads to maintain a high classification performance compared to using the typical vector representation.

Barbosa and Feng [11] proposed using microblogging features such as retweets, hashtags, replies, punctuations, and emoticons. They found that using these features to train the SVMs enhances the sentiment classification accuracy by 2.2% compared to SVMs trained from unigrams only. A similar finding was reported by Kouloumpis et al. [82]. They explored several types of microblogging features including emoticons, abbreviations and the presence of intensifiers such as all-caps (i.e., words with all characters in upper case) and character repetitions (e.g., `looove`) for Twitter sentiment classification. Their results show that the best performance comes from using n-grams together with microblogging features and the lexicon features, where words tagged with their prior polarity (positive, negative, neutral).[23] On the other hand, using POS features produced a drop in performance.

Agarwal et al. [1] also explored the POS features, the lexicon features and the microblogging features. Apart from simply combining various features, they also designed a tree representation of tweets to combine many categories of features in one succinct representation. A partial tree kernel [103] was used to calculate the similarity between two trees. They found that the most important features are those that combine prior polarity

23 Lexicon features refer to word n-grams (usually unigrams) coupled with prior sentiment in tweets. The prior sentiment of words is usually extracted using sentiment lexicons such as MPQA and SentiWordNet.

of words with their POS tags. All other features only play a marginal role. Furthermore, they also showed that combining unigrams with the best set of features outperforms the tree kernel-based model and gives about 4% absolute gain over a unigram baseline.

In addition to the work in, [82, 1] Bravo-Marquez et al. [22] have deeply investigated the usefulness of the lexicon features for sentiment analysis of tweets. In particular they proposed incorporating three types of lexicon features into classifier training: (i) words' prior polarities (positive, negative), (ii) emotions expressed in tweets (e.g., sadness, joy, surprise), and (ii) sentiment strength, a numerical value that refers to the intensity of a tweet's sentiment (see Section 2.1.1). These features were extracted using different lexicon resources and third-party sentiment analysis tools, including the OpinionFinder lexicon [176], the AFINN lexicon [106], SentiWordNet, the NRC lexicon [100], the SentiStrength algorithm [160] and the Sentiment140 service [60]. The authors also analysed the discriminative power of all the extracted features for both subjectivity and polarity classifications. They found that SentiWordNet, OpinionFinder and AFINN-based features are the most informative features for both classification tasks. They also showed that, while the prior polarity features are highly discriminative, the emotion features have almost no discrimination values for subjectivity classification. The evaluation was done using CART, J48, NB, Logistic Regression (LR), and SVM classifiers. Results showed that SVMs trained from a combination of the three type of features outperform all other combinations by more than 5% in accuracy and F-measure.

From the above, one can notice that although a wide range of features has been exploited for classifier training, the question remains of, which set of features or which combination of them is optimal for sentiment analysis on Twitter.

All Types of Features!

In June 2013, the *Semantic Evaluation* workshop (SemEval) organised a sentiment analysis task on Twitter [104].[24] To this end, the participants were provided with a manually labelled Twitter dataset, which allows for sentiment analysis classification at tweet and expression levels. Following the machine learning approach, several supervised models have been proposed and trained from several sets of features [101, 96, 127, 175, 79, 105, 122, 64, 94]. However, what was interesting about the challenge is that the winning approach [101], which produced the highest performance among other 44 approaches, was simply based on a supervised classifier trained from a combination of a very large

24 http://www.cs.york.ac.uk/semeval-2013/task2/

number of different types of features. Specifically, Mohammad et al. [101] used a SVM classifier trained from a combination of word n-grams (1,2,3 and 4-grams), character n-grams (3,4 and 5 characters), all-caps, POS tags, lexicon features, negation, hashtags, punctuations, clusters (the word's cluster) and emoticons. The lexicon features were extracted from three pre-built lexicons: NRC, MPQA, and Bing Liu [71] as well as from two lexicons constructed by the authors specifically for this task. The proposed model was evaluated on a set of 3,813 tweets for subjectivity classification, achieving the highest performance of 69.02% in macro F-measure when trained from all type of features. The authors also studied the contribution of each feature set to the classification performance by training the model from all the feature types but excluding one type at a time. Several interesting findings were reported. For example, unlike Barbosa and Feng [11] and Kouloumpis et al. [82], the authors found that the word and character n-grams are very important features to the classification performance, while microblogging features such as emoticons and hashtags have no actual impact on the performance.

It is also worth noting that other approaches that achieved good performance in the SemEval task followed a similar approach to [101], i.e., choosing a sentiment classifier and training it from all or subsets of the above features. Gunther and Furrer [175] proposed training a linear model from a collection of normalised tokens, stemmed words, words' clusters and lexicon features (based on SentiWordNet) using stochastic gradient descent (SGD). Proisl et al. [122] used SVM and NB trained from lexicons features, slangs and emoticons. Rodríguez-Penagos et al. [133] trained SVM and CRF classifiers from lexicon features, negation and quantifiers.

Rather than using a single classifier for classification, Kökciyan et al [79] employed a multiple classification models, where a NB classifier is used for negative sentiment classification while a MaxEnt classifier is used for positive sentiment classification. A rule-based system was then used to decide on the overall sentiment of a given tweet based on its negative and positive probabilistic scores obtained from each classifier. Features, such as microblogging features, repetitions (i.e, words with repeated letters), negation and lexicon features, were used for classifier training. The best performance of 63.53% was obtained using a combination of slangs, negations, sentiment words and ends-with-exclamation features.

3. Types of Machine Learning Classifiers

The third key factor to the effectiveness of the machine learning approaches is the choice of the classification algorithm or in other words the type of the sentiment classifier. In the previous subsections, we have seen that the most previous works used supervised classifiers including NB, SVM, MaxEnt, KNN, CRF, C4.5, CART, J48, LR, etc. [60, 116, 110, 11, 13, 45, 1, 82, 123, 101, 96, 127, 175, 79, 105, 122, 64, 94]. While, some works used standard versions of the classifiers [60, 11, 22, 101], others altered them in various ways to better suit the nature of tweets data [91, 110]. This further gave a greater variations of models of different performances. However, the lowest performance has been always much higher than the performance of random classifiers. With such a great variation, there is usually no ultimate or best-choice classifier. This is because the performance of the classifier is highly dependent on the training methodology, the type of features and the type of sentiment analysis task. Having said that, one can fairly argue that SVM, MaxEnt and NB are among the most popular classification methods used for sentiment analysis of tweets.

Summary

Based on the above review, we notice that the supervised machine learning approach has proven successful in sentiment classification on Twitter. Nonetheless, the social nature of Twitter, along with the special characteristics of tweets pose several limitations to this approach. Firstly, it relies on large annotated corpora of tweets for classifier training. Sentiment annotation of textual data is usually expensive [89], especially for continuously changing and evolving subject domains as on Twitter. Although some of the aforementioned studies relied on the the distant supervision approach for sentiment annotation of tweets, this approach has been criticised for harming the sentiment classification performance due to its relatively low annotation quality [152]. Secondly, supervised approaches are usually domain dependent, i.e., classifiers trained on data from one specific domain (e.g., tweets on political events) could produce low performance when applied to data from a different domain (e.g., tweets on social events) [6]. Thirdly, tweet messages tend to be very sparse due to the frequent use of abbreviations, ill-formed words and irregular expressions in tweets. The sparsity problem usually hinders the classification performance since many terms in the training data do not appear in the test data [137].

Some attempts have been made to overcome the above limitations, as discussed earlier. For example, semi-supervised classifiers were used to reduce reliance on automatically

labelled data or distant supervision [152], and different feature engineering processes and dimensionality reduction techniques were adopted to reduce the sparseness of tweets data [4, 5].

However, a common limitation that still face the above reviewed works is that they are **semantically weak** since they heavily rely on features extracted from the syntactic, linguistic or lexical representation of words for classifier training. With the absence of these features, which usually denote sentiment, the trained classifiers tend to fail in extracting the correct sentiment in tweets, as will be explained in Section 2.2.1.4.

Our work in this thesis targets the above limitation of machine learning approaches by extracting and using the latent semantics of words, which implicitly reflect sentiment in tweets, as features to train sentiment classifiers, as will be discussed in Chapters 4 and 5.

2.2.1.2 *The Lexicon-based Approach*

The lexicon-based approach to sentiment analysis presumes that the sentiment orientation of a given document can be inferred from the sentiment orientation of its words and phrases. Unlike the machine learning approach, the lexicon-based approach does not require feature engineering or classifier training. Instead, it uses sentiment lexicons to assign sentiment to the opinionated words in the document.

The efficiency of a lexicon-based sentiment analysis method is usually determined based on the type of the sentiment lexicon and the sentiment detection algorithm, i.e, the algorithm used to detect the opinionated words in the text and calculate the overall sentiment.

In Twitter sentiment analysis, several lexicon-based methods have recently been proposed [164, 9, 20, 23, 160, 72]. They can be categorised into (i) *Conventional Keyword Matching Methods* and (ii) *Twitter-based Methods*.

1. Conventional Keyword Matching Methods

Conventional keyword matching methods on Twitter (aka *conventional lexicon-based methods*) use a pre-built sentiment lexicon along with a simple keyword matching algorithm. Lexicons, such as SentiWordNet, MPQA and LIWC, are commonly used in this vein. For example, Tumasjan et al. [164] used the LIWC (Linguistic Inquiry and Word Count) software [117] to extract positive and negative emotions from a collection of 104,003 tweets published in the weeks leading up to the German federal election to predict election results. Instead of handling each tweet individually, tweets published over the relevant

time frame were concatenated into one text sample and are mapped into 12 emotional dimensions (e.g., anger, sadness, anxiety, etc.). Bae and Lee [9] also used LIWC, but this time for measuring the popular and influential users on Twitter based the sentiment of their tweets as well as the sentiment of tweets posted by their followers.

Conor et al. [108] calculated the sentiment polarity of a tweet by simply searching for polarity-bearing words (i.e., positive and negative words) within it. The MPQA subjectivity lexicon was used to this end. The tweet is considered positive if it contains any positive word, negative if it contains any negative word, and mixed if it contains both. Tweets were acquired from two different corpora: the The Obama job approval corpus[25] and the Obama-McCain Presidential Elections 2008 corpus.[26] In addition to extracting the sentiment of individual tweets, the authors also calculated the average sentiment of tweets published in a specific day (i.e., the per-day sentiment) by taking the ratio of positive to negative words found in the aggregated tweets. The sentiment detection performance was checked manually and the evaluation results showed a very low recall with many false classified samples. This was explained as tweets in both tweet corpora contain a large number of infrequent and ill-formed, yet opinionated words, which were not covered by the MPQA lexicon.

Bollen et al. [20] adopted a similar word counting approach using the MPQA lexicon to sentiment detection of tweets for stock market prediction. However, rather than using all the words in the lexicon, the authors exploited only those words of "*weak*" and "*strong*" sentiment (e.g., week positive, strong negative). In addition to binary sentiment detection, the authors also proposed a method to capture more refined emotional states in tweets such as "*Calm*", "*Alert*", "*Sure*", etc. To this end, they built a new lexicon of 964 words coupled with their emotion labels. The new lexicon was built upon the POMS lexicon [107] and expanded by adding 892 additional new words to the 72 words in the original lexicon. The results showed that the inclusion of emotions led to increasing the performance of the predication model compared to using MPQA solely.

From the above, conventional lexicon-based methods have the advantage of simplicity, which becomes a requirement when analysing large corpora of tens of millions of tweets as in [108, 20]. However, these methods often face two main limitations:

25 http://www.gallup.com/poll/113980/Gallup-Daily-Obama-Job-Approval.aspx
26 http://www.pollster.com/polls/us/08-us-pres-ge-mvo.php

1. Conventional methods are typically poor in detecting complex forms of sentiment in tweets [160]. This includes, for example, negated sentiment (e.g., "`For some reason I` **`never`** `liked that song`") and boosted sentiment (e.g., "`I'm` **`very`** `happy for you man!`" or "`I feel` **`extremely`** `tired today`").

2. Traditional lexicons such as MPQA and SentiWordNet are designed to work with formal and well-written English text. In Twitter, however, the use of malformed words and irregular expressions is rather popular due to the 140-character limit. This consequently leads to low retrieval of irregular opinionated words in tweets, since such words are not usually covered by traditional lexicons, as demonstrated in [108, 23].

2. Twitter-based Methods

Twitter- or microblogs-based methods for lexicon-based sentiment analysis are more tailored to tweets data (and the like) than conventional lexicon-based methods, and therefore, they tend to obtain better sentiment detection performance. This is because Twitter-based methods try to overcome the aforementioned limitations by: (i) bootstrapping existing lexicons, or creating new lexicons designed to work specifically on tweets data [159, 106], and (ii) building algorithms that take into consideration the special characteristics of tweets when identifying and extracting their sentiment [23, 160, 72].

Pioneering work in this vein is by Thelwall et al. [159], who built SentiStrength, a lexicon-based method for sentiment strength detection on microblogging data. In the approach of Thelwall and colleagues, the sentiment strength of a given post is a numerical value representing the intensity of the post's sentiment. Specifically, the negative sentiment strength is taken to be a number between -1 (not negative) to -5 (extremely negative). Similarly, the positive sentiment strength is a number between 1 (not positive) and 5 (extremely positive). SentiStregnth uses a human-coded lexicon of words and phrases specifically built to work with social data. We refer to this lexicon in this thesis as *Thelwall-Lexicon*. The lexicon initially consisted of 298 positive terms and 465 negative terms extracted from a collection of MySpace comments and manually annotated with sentiment strengths values between 2 and 5. A training algorithm was proposed subsequently to optimize the terms' sentiment strengths. In particular, the algorithm iteratively selects a term, updates its sentiment strength by -1 or +1, and checks whether that improves the classification performance on a corpus of human annotated texts. To overcome the problem of ill-formed language, SentiStrength applies several lexical rules, such as the ex-

istence of emoticons, intensifiers, negation and booster words (e.g., absolutely, extremely). To this end, the authors built three lists of booster, emoticons and negation words. The assessment of SentiStrength was done by comparing its performance against random and majority class classifiers as well as several machine learning classifiers including NB, SVM and J48. To this end, the authors used a test set of 1,041 MySpace[27] comments. The comments were manually annotated with their sentiment strengths by three human coders. Evaluation results showed that SentiStrength significantly outperformed the best machine learning classifier by over 2% for the positive sentiment strength detection. The authors claimed that the gain in performance is due to the ability of SentiStrength in dealing with the irregular English forms in text and using them to boosting the overall detection performance.

In a subsequent work [160] by the same authors, several improvements were added to the original algorithm. This mainly included extending the Thelwall-Lexicon to 2,310 terms by adding the negative terms of the General Inquirer lexicon [153]. The authors also updated the negation rule, where negating the negative terms in the new version of SentiStrength turns them neutral rather than positive. The new SentiStrength was evaluated on 6 test datasets of different social media services including Twitter, BBC Forum, Digg.com, MySpace, Runners World Forum and YouTube. The results showed that SentiStrength outperformed the baselines method (majority classification) over the six datasets significantly. However, the machine learning methods were shown to perform slightly better than SentiStrength on most of the datasets.

The same authors [111] proposed another approach based on SentiStrength, where the focus was toward polarity and subjectivity detection rather than sentiment strength detection. In particular, the authors used similar sentiment rules to those used in SentiStrength to calculate the positive and negative sentiment strength of a given document. The strength scores were subsequently used to compute the document's overall sentiment. The document is considered positive if its positive strength is larger than its negative one and vice versa. Moreover, the document is considered neutral (objective) if the absolute values of its positive and negative strengths are equal to 1. The proposed algorithm was evaluated on three human-annotated datasets collected from Twitter [110], Digg [112] and MySpace [158], and its performance was compared against SVM, NB and MaxEnt classifiers trained from word unigrams. The evaluation results showed a superior performance of the algorithm over all other methods for both polarity and subjectivity classification.

27 https://myspace.com/

From the above review, one can notice that adjusting lexicon-based algorithms to Twitter characteristics, as well as using microblogging-based lexicons help in reducing the negative impact of language informality in tweets, and consequently improving the detection performance, as also argued by [23]. Nonetheless, building sentiment lexicons is usually a time-consuming and labour-intensive task [89]. Moreover, similar to the traditional lexicons, the coverage of microblogging-based lexicons is also limited by the number of words within them.

The difficulty of building sentiment lexicons for microblogs, as explained above, attracted several research works that mainly focused on investigating (i) new types of opinionated words and expressions found in microblogging texts, and (ii) which of these word types are more useful to consider when constructing sentiment lexicons [106, 72, 23].

For example, Nielsen [106] studied the influence of including ill-formed expressions (e.g., `lol`, `gr8`) and obscene terms [140] that convey sentiment when building new sentiment lexicons. In particular, the authors built a new lexicon of 2477 words coupled with their sentiment strength scores (between -5 and +5). The words were collected from different sources including a list of obscene words [21], the *Original Balanced Affective Word List*,[28] a list of a slang words from the Urban dictionary,[29] etc.[30] Using a simple keyword matching, the new lexicon was compared to the ANEW [21] lexicon, General Inquirer, OpinionFinder and SentiStrength on a dataset of 1,000 tweets manually labelled with Amazon Mechanical Turk (AMT) [15]. The evaluation results showed that the new lexicon showed a higher correlation with the sentiment labels obtained from AMT than ANEW and General Inquirer and OpinionFinder, but a lower correlation than SentiStrength. This is because SentiStrength, as described earlier, applies several syntactical rules besides covering slangs and other ill-formed words and irregular expressions.

One way to overcome the limited coverage of words is by lowering the reliance on the sentiment lexicons, and more specifically on the reliance on the prior sentiment of words, and focus more on other sentiment indicators in tweets. For example, Hu et al [72] exploited *emotional signals*, i.e., these parts in text that convey sentiment or correlated to other parts that convey sentiment, for sentiment detection of tweets in an unsupervised manner. As defined by the authors, these signals are: (i) the words' prior sentiment, (ii) sentiment indicators in tweets (e.g., emoticons), (iii) word-word sentiment correlation (i.e., the correlation between opinionated and ordinary words in a given tweet), and (iv)

28 http://www.sci.sdsu.edu/CAL/wordlist/origwordlist.html
29 http://www.urbandictionary.com
30 The complete list can be found in the author's work in [106].

post-post sentiment correlation, which refer to the correlation between opinionated and ordinary posts. The authors proposed a matrix factorisation based framework [48] where the emotional signals are used as prior knowledge to constrain the factorisation process. Evaluation results on two Twitter datasets showed the effectiveness of the proposed approach in comparison with other conventional keyword matching methods.

In a similar vein, Brody and Diakopoulos [23] explored including *lengthening words* (i.e., words with repeated letters) as additional opinionated words to the lexicon. In particular, the authors argued that words such as "*reallllly*" and "*niiiice*" tend to have a strong association with sentiment, but usually are not covered by typical lexicons. Using MPQA, they found that 7% of the words in the lexicon appeared lengthened in 68% of the tweets in a corpus of 500K tweets. Based on this finding, the authors proposed to expand the MPQA lexicon with a subset of 720 lengthening words found in tweets. To this end, they ran a label propagation method on a graph of the lengthening words to calculate their sentiment. The annotation performance of the method was compared to a human annotation on two sets of 50 positive and 50 negative words. The overall results showed that word lengthening is not random and does often denote sentiment in the text.

Summary

From the above review, lexicon-based methods seem like a better fit for Twitter than machine learning methods since the former can be directly applied to tweets with no need for feature engineering and classifier training. Nonetheless, lexicon-based methods are limited by their lexicons. Firstly, they rely on sentiment lexicons of a fixed set of words. Words that do not appear in the lexicon are often not considered when analysing sentiment. Moreover, lexicon-based methods, and their underlying lexicons, offer fixed, context-independent, word-sentiment orientations and strengths. For example, typical lexicon-based methods assign the same sentiment strength to the word "`good`" in "`It is a very good phone indeed!`" and in "`I will leave you for good this time!`". Although training algorithms have been built to optimise the terms' sentiment score in the lexicon, as in [159], they require frequent retraining from human-coded data, which is labour-intensive and domain dependent.

The work in this thesis aims to address the above limitations by building a lexicon-based approach that considers the context of words, by means of their co-occurrence patterns in tweets, and calculates their sentiment orientation and sentiment strength accordingly. In doing so, our approach is able to find new opinionated words in tweets,

and uses them to extend general-purpose sentiment lexicons that do not normally cover such words, as will be discussed in Chapter 3.

2.2.1.3 *The Hybrid Approach*

The *hybrid approach* to sentiment analysis combines both, the machine learning and lexicon-based approaches, in a way that exploits the strengths of both approaches, but avoids some of their weaknesses. Broadly speaking, in the hybrid approach, one of the two approaches is used to boost the performance of the other approach. For example, supervised classifier models make use of lexicon-based approaches to lower the dependency on manually annotated training corpora [184]. On the other hand, the machine learning approaches can be used to bootstrap the sentiment lexicons used in the lexicon-based approaches [159].

A very popular form of hybridisation between both approaches is to incorporate the prior sentiment information of words as additional features into supervised classifier training as in [82, 1, 22, 101] (see Section 2.2.1.1). Such incorporation has been shown to improve the overall performance of sentiment classifiers [22].

Other existing works developed more sophisticated forms of hybridisation [184, 62, 128, 133]. For example, Zhang et al.[184] devised a three-step incremental learning approach. In the first step a lexicon-based method is applied to a collection of tweets to identify their sentiment labels (positive and negative). In the second step, the labelled tweets are used to identify the opinionated terms, which were not detected in the first step, based on their co-occurrence with positive and negative tweets. The extracted terms in this step are subsequently added to the lexicon used in the first step and used to find more opinionated tweets. In a further step, a SVM classifier is trained from the annotated tweets in first step and applied to classify the sentiment orientation of the named entities in the tweets detected in the second step.

Rodriguez-Penagos et al. [133] built an ensemble approach that uses the outputs of a lexicon-based method and two supervised classifiers to perform sentiment detection at the expression level. In particular, the authors trained SVM and CRF classifiers from several sets of lexicon features (e.g., prior polarity, polarity strength, subjectivity clue, etc). They also built a lexicon-based method that uses some pre-defined heuristic rules (e.g., if [negation (word)] then [flip sentiment]) along with several polarity lexicons. Majority voting and Ensemble voting (i.e., choose the output that maximise the number of correct

identified samples in the development corpus) strategies were used to calculate the total sentiment from the three outputs.

Revathy and Sathiyabhama [128] proposed using three supervised models trained from three different set of features: The first model is a random forest classifier trained based on compositional semantic rules [39].[31] The second model is a SVM classifier trained from the senses of words extracted WordNet [98]. The third model uses our previously proposed semantic features (e.g., "`Person`", "`Company`", "`Person`") (see Chapter 4) to train a Naive Bayes classifier [136]. The final sentiment of the tweet corresponds to majority votes by the three models. The evaluation results showed that the proposed approach outperform other supervised classifiers trained from several combination of word n-grams (unigrams, bigrams) and POS tags. It also outperform other models trained from the three semantic features solely.

Gonçalves et al. [62] proposed an approach that combines between several lexicon-based methods including SentiStrength, Emoticons (the presence of emoticons in tweets as an indicator to their sentiment) SenticNet (Keyword matching against the SenticNet lexicon) [28] PANAS-t [63],(A lexicon-based method built based on the psychological positive affect and negative affect scale (PANAS) [172]) SentiWordNet (Keyword matching against the SentiWordNet lexicon), Happiness Index (Uses the ANEW lexicon to measure the happiness index (from 1 to 9) in text) [50], as well as SASA [170], a supervised method that uses a Naive Bayes classifier trained from 17K manually annotated tweets for subjectivity classification. The sentiment produced by the combined method is calculated as the harmonic mean of the precision and recall of all the methods. The combined method along with each of the aforementioned methods were evaluated using the 6 social datasets from [160]. Evaluation results showed that the combined method outperform all the other methods in terms of coverage (i.e., the proportion of tweets that the method is able to detect their sentiment) and agreement (i.e., the proportion of tweets that both the method and the ground truth agreed on their sentiment).

2.2.1.4 *Discussion*

In the preceding sections we provided an overview of two popular approaches to sentiment analysis on Twitter, the machine learning approach and the lexicon-based approach.

31 A set of linguistic rules that are used to determine the sentiment of the a tweet as a composition of the the sentiment of the expressions within the tweet.

Both approaches have quite distinct characteristics that lead to several advantages and disadvantages when applied to Twitter data.

In this section we briefly summarise and discuss the strengths and weaknesses of both approaches as follows:

Supervised Machine Learning Approaches

Traditional machine learning approaches are domain-dependent, which can be a leverage and a weaknesses at the same time. On the one hand, classifiers trained from a specific domain tend to produce high performances when applied to data from the same domain. This is because the training process ensures that the trained classifiers are well-adapted to the domain, topic and context of the training data. On the other hand, classifier training requires a large amount of annotated data, which is not always available especially in Twitter. Although the distant supervision approach to building training corpora has been adopted in several works, its efficiency is still inconsistent, as discussed in Section 2.2.1.1. Moreover, domain-dependent classifiers usually fail to produce a satisfactory performance when applied to new domains; classifier retraining is often needed in such cases. Therefore, the efficiency of the supervised machine learning approach is usually subject to two conditions: one is the availability of annotated corpora, another is classifier retraining when move to corpora of different domains.

Another common characteristic of traditional machine learning approaches on Twitter is that they merely rely on syntactic or linguistic features for classifier training (e.g., n-grams, POS tags, dictionary glosses, etc.). This is sometimes problematic on Twitter because relying on these types of features lead to generate sparse feature vectors since tweets are naturally sparse, as discussed earlier. Using sparse vectors to train sentiment classifiers often lower their performance [137] due to their inability to generalise to new terms found in the test set, which the classifiers are not trained from (or trained from their associated feature vectors).

Lexicon-Based Approaches

Lexicon-based approaches often rely on sentiment lexicons for sentiment analysis and, therefore, they do not require training from labelled instances or retraining when applied to Twitter corpora of different domains, which is generally counted as a strength for some sentiment analysis applications on Twitter (e.g., analysing sentiment on Twitter streams). However, these approaches, as reviewed previously, are limited by the sentiment lexicons they use. First, because sentiment lexicons are composed by a generally

static set of words that could not cover the wide variety of new terms that constantly emerge in Twitter. Secondly, because words in the lexicons have fixed prior sentiment orientations, i.e. each term has always the same associated sentiment orientation regardless of the context in which the term is used.

The lack of Semantics

A common limitations with both, traditional machine learning and lexicon-based approaches to Twitter sentiment analysis is their full dependence on the presence of affect words or syntactic features that are explicitly associated with sentiment. However, it is very likely that the sentiment in a tweet is implicitly expressed via the semantics of its words or the semantic relations between them [24, 55, 126]. As previously explained in Chapter 1, the sentiment of a word is often associated with its semantics in a given context. Changing the context may lead to alter the word's semantics and consequently sentiment. For example, the word "`great`" should be negative in the context of a "`problem`", and positive in the context of a "`smile`". Moreover, the sentiment might be also associated with the explicit semantic concepts of words. For example, `Trojan Horse` is likely to have a negative sentiment when its associated semantic concept is "`Computer/Virus`". and it is likely to have a neutral sentiment when its associated semantic concept is "`Greek Mythology`". Understanding the semantic of words, therefore, becomes crucial for sentiment analysis for both, words and tweets.

The role of semantics in sentiment analysis has been increasingly investigated in different works in the last three years. We conventionally refer to this line of work as *Semantic Sentiment Analysis*. In the next section we provide a review of the exiting approaches in this line of work.

2.3 SEMANTIC SENTIMENT ANALYSIS

Semantic sentiment analysis aims at extracting and using the underlying semantics of words in identifying their sentiment orientation with regards to their context in the text. Existing works on semantic sentiment analysis could be divided into two main approaches: (i) the *contextual semantic approach* (aka the statistical semantic approach) and the (ii) the *conceptual semantic approach*. In the following subsections, we will review key existing works under each semantic approach, highlighting their properties, strengths and weaknesses.

2.3.1 Contextual Semantics

Contextual semantics (aka statistical or distributional semantics) refer to the type of semantics which is inferred from the co-occurrence patterns of words [177]. Contextual semantics have been traditionally used in diverse areas of computer science including Natural Language Processing and Information Retrieval [167]. The main principle behind the notion of contextual semantics comes from the dictum-"*You shall know a word by the company it keeps!*" [52]. This principle in Linguistics is also known as the *Distributional Hypothesis*, which states that words that occur in the same contexts tend to have similar meanings [65, 167, 134], which suggests that words that co-occur in a given context tend to have a certain relation or semantic influence.

In the sentiment analysis area, the above principle has been adopted in several approaches in order to assign sentiment to words based on their co-occurrence patterns in text. [67, 165, 166, 157, 171, 87, 88, 179]. Similar to traditional sentiment analysis approaches, contextual semantics approaches can be divided into two main categories: lexicon-based approaches [165, 166, 157, 87, 88] and supervised approaches [171, 179].

Sentiment Orientation by Association.[32] (SOA) [165, 166, 157] is a popular approach under the lexicon-based category.[33] SOA assigns sentiment orientation to words based on the sentiment orientation of their co-occurring words in a text [165, 166], in dictionary glosses [157], or in a synonymy graph [78]. Pioneering work in this vein is by Turney and Littman [165, 166]. In their work, the authors exploited the statistical correlation between a given word and a balanced set of 14 positive and negative paradigm words (e.g., good,

32 Also called *Semantic* Orientation by Association

33 SOA is considered an unsupervised approach in some studies.

nice, nasty, poor). The word has positive orientation if it has a stronger degree of association to positive words than to negative ones, and vice-versa. Two measures of word statistical correlation were used, *pointwise mutual information* (PMI) [40] and *Latent Semantic Analysis* (LSA) [86].[34] The evaluation was conducted on 3596 words obtained from General Inquirer. Although this work does not require large lexical input knowledge, its identification speed is very limited [179] because it uses web search engines in order to retrieve the relative co-occurrence frequencies of words.

Other forms of semantic associations can be realised at more abstract levels than the corpus level. For example, Takamura et al. [157] measured the the sentiment orientation of words based on their association to seed words in dictionary glosses. In particular, they built a network of terms where two terms are connected if one appears in the gloss set of the other. After that, a spin model [37] was applied to decide on the total sentiment of terms in the network. All word glosses were obtained from WordNet. In a similar vein, Kamps et al. [78] represented the semantic association between words as a graph of WordNet synonymy relations. The sentiment of a word in their approach corresponds to the shortest path of two possible paths, one from the word to the seed word "`good`", and another to the seed word "`bad`".

Although using lexical inputs from other resources (e.g., glosses and synonyms from WordNet) is more time-efficient than using the Web, the aforementioned approaches are limited by their lexical resources. For example, the proposed approach in [78] assigns sentiment only to those words that appear in WordNet.

SOA approaches, in general, are unable to assign sentiment to words with domain- and context-specific orientations [49] due to (i) their limited choices of paradigm words, which are usually general and out-of-context, and (ii) their use of external lexical data sources (Web, WordNet, etc) that do not reflect, most of the time, the contextual semantics of the words in the studied corpus. For example, SOA approaches are generally unable to distinguish between "`Heavy`" as a negative word when describing mobile phone and as a positive word when describing a wood dining table.

Wang and Wan [171] proposed a supervised approach to contextual semantics for sentiment analysis of online news documents. Specifically, the authors used LSA to lower the dimensionality of passage-word semantic space and extract the statistical correlation

34 LSA is a statistical semantic technique that uses Singular Value Decomposition (SVD) to extract the latent co-occurrence patterns of words in text. It can be thought of as a data smoothing technique that aims at reducing the dimensionality of the data in order to decrease the distances between the vectors of semantically similar words or documents in the semantic space.

between words in different passages (In other words the latent semantics of the passage). After that, a sentiment lexicon was used to assign prior polarities to the words in the smoothed space and the sentiment of documents were calculated consequently as an average sum of the their opinionated words. In a further step, the authors trained a SVM classifier from the labelled documents. The authors claimed that their model outperformed another model that followed similar steps but without using SVM classifiers.

Latent Dirichlet Allocation (LDA) [19] is a probabilistic generative model, which presumes that the document is a mixture of probabilities of topics and each topic is a mixture of probabilities of words that are more likely to co-occur together under the topic. For example the topic "`Flu`" is more likely to contain words like "`headache`" and "`fever`" while the topic "`iPhone`" is more likely to contain words like "`display`" and "`battery`". Hence, one can interpret the topics generated by LDA as the contextual semantic identities of words since the words under each topic tend to have some sort of semantic relations between them.

Several extensions to the original LDA model have been proposed to perform different sentiment analysis tasks including subjectivity detection [87, 88], aspect-level sentiment detection [186, 162], topic-based sentiment summarization and time-series sentiment analysis [70].

For example, Lin and He [87] proposed the joint sentiment-topic (JST) model, a four-layer generative model which allows the detection of both sentiment and topic simultaneously from text. The generative procedure under JST boils down to three stages. First, one chooses a sentiment label l from the per-document sentiment distribution π_d. Following that, one chooses a topic z from the topic distribution $\theta_{d,l}$, where $\theta_{d,l}$ is conditioned on the sampled sentiment label l. Finally, one draws a word w_i from the per-corpus word distribution $\phi_{l,z}$ conditioned on both topic z and sentiment label l. The graphical model of JST is presented in Figure 6.[35] The JST model does not require labelled documents for training. The only supervision is word prior polarity information which can be obtained from publicly available sentiment lexicons such as the MPQA subjectivity lexicon.

Unlike SOA approaches, LDA-based approaches derive the contextual semantics of words from their statistical correlations to other words that co-occur with them in the corpus being analysed. Therefore, the LDA-based models better captures the words' specific context and consequently their contextual sentiment orientations. Having said that, LDA-based models face several challenges. First, they tend to a have high computational

35 The figure is taken from the authors' work in [87] with permission.

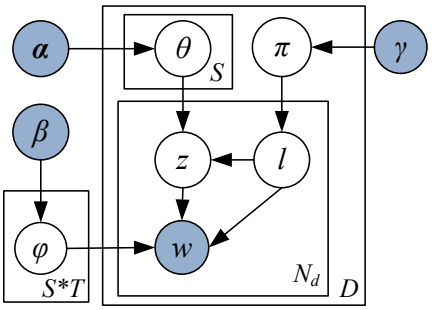


Figure 6: JST Model [87].

complexity [151] because the MAP (maximum a posteriori) inference (i.e., $P(\mathbf{z}|\mathbf{w})$) in these models is an NP-hard problem. Although several approximate inference algorithms have been proposed to increase (e.g., Gibbs sampling, Markov Chain Monte Carlo) the time-efficiency, the performance of an LDA-based model is highly dependent on the number of topics per documents, which is usually chosen heuristically. Secondly, LDA-based models often rely on the bag-of-word representation of documents, where the syntactical structure of sentences/phrases is usually ignored. Hence, the sentiment prediction in such models tends to fail in simple cases such as the presence of negation.

2.3.2 Conceptual Semantics

The conceptual semantic approach to sentiment analysis relies on external semantic knowledge bases (e.g, ontologies and semantic networks) with NLP techniques to capture the conceptual representations of words that implicitly convey sentiment. Unlike contextual semantic approaches, which are mainly based on the bag-of-word paradigm, the conceptual semantic approach relies on the bag-of-concept paradigm, that is, by representing the affective (sentiment) knowledge in text as a set of concepts coupled with their sentiment orientations [26].

Conceptual semantic approaches are relatively new in the sentiment analysis and less popular than contextual semantic approaches. Pioneering work in this vein is by Cambria and Hussein [25], who introduced *Sentic Computing*,[36] a new paradigm for concept-level sentiment analysis. Sentic computing employs two knowledge-base sources of com-

36 The term is derived from the Latin *sentire*, which is also the root of words like *sentiment* and *sentience*.

mon sense concepts and affective information, ConceptNet[37] [68] and WordNet-Affect[38] (WNA) [155] along with common sense reasoning techniques. These resources and techniques are used to extract the common-sense concepts (e.g., "`nice day`", "`simple life`") found in text with their associated semantics and sentics (i.e., four affective categories derived from the Hourglass of Emotions model [32]: *Pleasantness, Attention, Sensitivity and Aptitude*). In particular, the sentiment detection in the new paradigm boils down into three main phases as shown in Figure 7. First, a semantic parser[39] is used to extract common sense concepts from a given text. Secondly, the extracted concepts are passed to the semantic extraction module. This module extracts the semantic classes of the extracted concepts based on their relatedness to a predefined set of seed concepts in the common sense semantic network, *IsaCore* [34].[40] Third, the extracted concepts are also passed to the sentic extraction module. In this module each concept is assigned a vector of four sentic components (pleasantness, attention, sensitivity, aptitude) based on its position in the *AffectiveSpace*, a multi-dimensional vector space model in which the concepts in ConceptNet (common sense concepts) are represented/clustered by means of their affect information in WordNet-Affect.

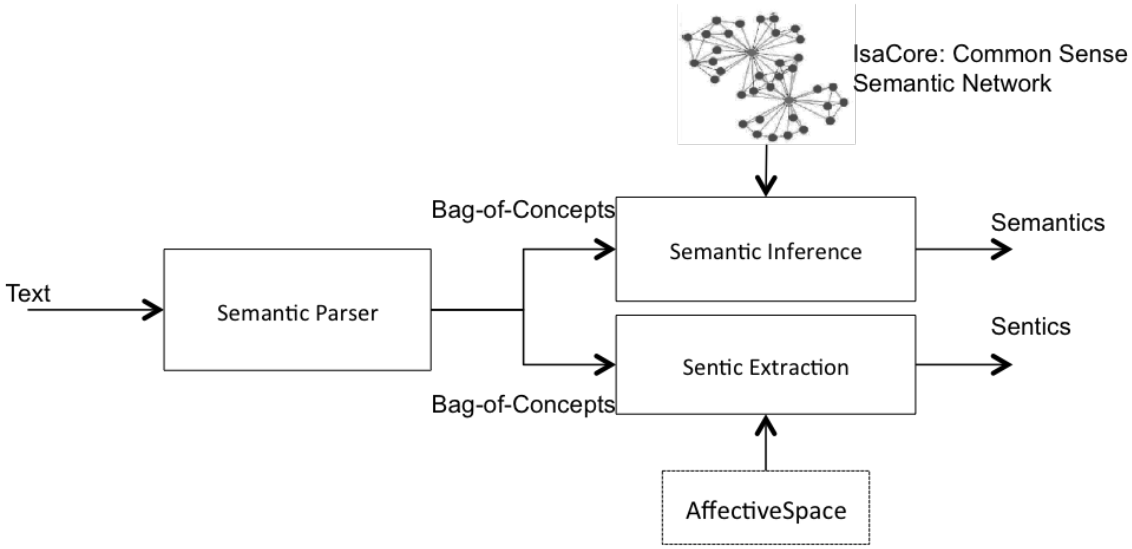


Figure 7: Concise paradigm of Sentic Computing. The detailed paradigm can be found in [25].

37 ConceptNet is a semantic graph representation of the common sense information in the Open Mind corpus. Nodes in the graph represents concepts and edges represents the assertions of common-sense that interconnect concepts.

38 WordNet-Affect is a linguistic resource of affective knowledge built upon WordNet.

39 `http://sentic.net/parser.zip`

40 IsaCore is a semantic network representation of common and common-sense knowledge, built by blending Probase [178] and ConceptNet. IsaCore can be downloaded from `http://sentic.net/isacore.zip`

Note that AffectiveSpace is built using the blending technique over ConceptNet and WordNet-Affect. In particular, the common sense knowledge in ConceptNet and the affective lexical information in WordNet-Affect are represented as two sparse matrices. Blending exploits the shared information between both matrices in order to merge them into a single matrix. After that, a truncated singular value decomposition is applied in order to lower the dimensionality of the blended matrix and captures the latent correlations between concepts. As a final result, common sense concepts like such as "`happy birthday`" and "`birthday present`" are jointly represented with their affective information as neighbour vectors in the AffectiveSpace.

Several data types were used to evaluate the different sentic computing components. For example, AffectNet was evaluated for emotion detection on a set of 5000 blog posts from LiveJournal [154],[41] while the ensemble of IsaCore and AffectNet was evaluated for polarity detection on a set of 2000 tagged comments obtained from PatientOpinion,[42] an online feedback service for the National Health Services (NHS) in the UK.

As for Twitter, only the semantic component was evaluated. In particular, a corpus of 3000 tweets from [150] was used to evaluate IsaCore. The hashtags in this dataset were used to tag categories and topics in tweets such as companies (e.g., Apple, Microsoft, Google), cities, countries, etc. The evaluation showed that IsaCore outperform significantly both WordNet and ConceptNet. It also outperformed Probase [178] in some categories (e.g., electronics, cars). However, the tweets in the evaluation corpus, as explained in [25] contain formal, structured and well-written English text. Additionally, the common sense concepts are well defined and represented in the tweets with at least one concept in each tweet. Hence, the evaluation corpus does not reflect the typical characteristics of Twitter data (e.g., malformed words and unstructured sentences), and therefore, the performance of IsaCore on tweets data is yet to be measured using larger and more realistic Twitter corpora.

It is worth noting that sentic computing as a paradigm was applied effectively in several subsequent works by Cambria and his colleagues [27, 38, 29, 30] to build sentiment analysis applications for different fields such social media marketing [30] and health care services [29]. In this line of work, sentic computing has proven its effectiveness for sentiment analysis at the concept, aspect, and document levels. Nevertheless, sentic computing approaches face quite a few limitations. Firstly, the main component in sentic

41 LiveJournal is a social network that allows users to publish and share blogs, jorunals and diaries http://www.livejournal.com/

42 https://www.patientopinion.org.uk/

computing, the AffectiveSpace, is built using the SVD reduction technique on a very large and sparse blended matrix. Such process is computationally expensive and requires big storage. This might become very problematic in Twitter, where the knowledge sources in sentic computing are supposed to keep extending with the continuous evolution of new concepts in tweets. Secondly, current sentic computing approaches rely on static common sense knowledge sources. Therefore, the performance of such approaches is affected, to a large extent, by the richness of these sources (i.e., the number and the diversity of the common and common-sense concepts that the knowledge source covers).

SenticNet

One way to lower the complexity of the sentic computing paradigm, is to generate the AffectiveSpace for once and store its common sense concepts along with their sentics in a static lexicon. This allows to extract the sentic information of a given common sense concept without the need for regenerating the AffectiveSpace. To this end, Cambria et al. [28] built *SenticNet*,[43] a concept-based lexicon for sentiment analysis. To build SenticNet, the authors first created the AffectiveSpace from ConceptNet and WordNet-Affect as described earlier. After that, a subset of 5700 concepts were selected from those which have positive or negative polarity only. The sentiment polarity of a given concept c was calculated from its 4 sentic classes scores as:

$$polarity(c) = \frac{Pleasantness(c) + |Attention(c)| - |Sensitivity(c)| + Aptitude(c)}{9} \quad (2)$$

SenticNet was further extended in [31] by adding all the polar concepts from the Open Mind corpus, resulting in 14k concepts coupled with their affective information.[44] Both versions of SenticNet were published in semantic web based format (RDF/XML format), where the semantics and the sentics of a concept like "`minor injury`" are represented in SenticNet as illustrated in Figure 8.

To evaluate SenticNet, the authors compared its performance against the traditional SentiWordNet lexicon using the PatientOpinion dataset. The results showed that SenticNet outperformed SentiWordNet with a significant gain of 18% in F-measure.

SenticNet has been recently used in a number of studies to perform different sentiment analysis tasks including aspect-level sentiment analysis of products [56], as well as, sentence-level and opinion holder sentiment detection [55].

43 http://sentic.net/senticnet-1.0.zip
44 http://sentic.net/senticnet-2.0.zip

```
<rdf:Description rdf:about="http://sentic.net/api/en/concept/minor_injury">
 <rdf:type rdf:resource="http://sentic.net/api/concept"/>
 <text xmlns="http://sentic.net/api/">minor injury</text>
 <semantics xmlns="http://sentic.net/api/" rdf:resource="http://sentic.net/api/en/concept/knee_injury"/>
 <semantics xmlns="http://sentic.net/api/" rdf:resource="http://sentic.net/api/en/concept/may_injure"/>
 <semantics xmlns="http://sentic.net/api/" rdf:resource="http://sentic.net/api/en/concept/score_goal"/>
 <semantics xmlns="http://sentic.net/api/" rdf:resource="http://sentic.net/api/en/concept/score_point"/>
 <semantics xmlns="http://sentic.net/api/" rdf:resource="http://sentic.net/api/en/concept/get_injure"/>
 <pleasantness xmlns="http://sentic.net/api/" rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0</pleasantness>
 <attention xmlns="http://sentic.net/api/" rdf:datatype="http://www.w3.org/2001/XMLSchema#float">-0.125</attention>
 <sensitivity xmlns="http://sentic.net/api/" rdf:datatype="http://www.w3.org/2001/XMLSchema#float">+0.086</sensitivity>
 <aptitude xmlns="http://sentic.net/api/" rdf:datatype="http://www.w3.org/2001/XMLSchema#float">-0.171</aptitude>
 <polarity xmlns="http://sentic.net/api/" rdf:datatype="http://www.w3.org/2001/XMLSchema#float">-0.044</polarity>
</rdf:Description>
```

Figure 8: The RDF entry of the concept `minor injury` in the SenticNet lexicon.

For example, Gangemi et al. [55, 126] proposed *Sentilo*, a frame-based semantic model for sentence-level sentiment detection. The proposed model is able to (i) detect topics and opinion holders (i.e., entities that hold an opinion) in sentences and (i) defines the sentiment expressed by an opinion holder towards a given topic. To this end, the model performs extensive syntactical, lexical and semantic analyses of sentences. Concisely, the sentiment detection in Sentilo consists of two steps. In the first step, a sentence like "`Anna says the weather will become beautiful`", is semantically represented as a sequence of interconnected events (e.g., `the weather will become beautiful`) and objects/agents (e.g., `Anna, Weather`). To this end, the authors used Fred,[45] a tool for semantic knowledge extraction and representation of sentences [121]. In the second step, the semantic representation of the sentence is used along with the SenticNet and SentiWordNet lexicons to detect the opinionated topics, subtopics, opinion holders and their sentiment. Sentilo was evaluated on 50 manually annotated sentences from Europarl corpus,[46] with results showing F-measures of 95%, 68%, 78% for opinion holder, topic and subtopic detection respectively.

Conceptual Semantics on Twitter

Based on the above review, one may notice that most existing conceptual semantic approaches to sentiment analysis were mainly applied and tested on conventional text. As for Twitter, however, very few attempts of exploiting conceptual semantics for sentiment detection of tweets can be found [57, 62, 80, 102].

Gezici et al, [57] used SenticNet as an ordinary sentiment lexicon along with SentiWordNet to extract the prior sentiment of words and use them to train a supervised regression classifier for sentiment classification of Tweets. However, no concept detection nor semantic inferencing was performed to this end.

45 http://stlab.istc.cnr.it/stlab/FRED

46 European Parliament Proceedings Parallel Corpus: http://www.statmt.org/europarl/

Kontopoulos et al. [80] proposed an ontology-based approach to sentiment detection of product aspects on Twitter. To this end, the authors built a domain-specific ontology for smart phones by first retrieving tweets that contain specific concepts (e.g., "`#smartphone`") in a predefined list. An ontology engineer was asked, afterwords, to manually extract the objects (e.g., "`Apple iPhone`", "`Samsung Galaxy`") and their attributes (e.g., "`Display`", "`Battery`") from the tweets. The evaluation was done by querying Twitter with the object-attribute pairs in the ontology. The sentiment of tweets with regards to the attributes in them (aspects) was calculated using OpenDover, an online commercial tool for sentiment analysis.[47] Although the proposed approach showed a relatively high recall, its main limitation comes from the high levels of intervention by human experts when building or validating ontologies. In practical applications on Twitter, building an ontology for each product is unscalable and too expensive.

Another recent work in this vein is by Montejo-Ráez et al. [102], who conceptualised words in tweets by means of their synsets[48] in Wordnet. In particular, word sense disambiguation was performed on words in tweets using the UKB method [2] in order to assign a unique synset to each word. After that, the authors applied a random walk process over Wordnet to expand the original list of sunsets. Finally, The polarity of a tweet is then calculated as an average of the polarity of all of its synsets. SentiWordNet was used to this end. Evaluation was conducted on 359 tweets from the STS corpus [60]. Results showed that weighting the prior sentiment of words in SentiWordNet using their extracted synsets did improve the overall performance by up to 23.12% in F1 over using SentiWordNet with most-common-sense weighting scheme. However, comparable to other supervised sentiment analysis approaches, including [60, 16, 152, 137], the performance of the proposed approach was 19% lower in F1, on average.

47 http://opendover.nl/

48 Sets of cognitive synonyms.

2.4 SUMMARY AND DISCUSSION

Sentiment analysis of Twitter is increasingly moving into the focal interests of several research and commercial parties. Twitter, however, poses several challenges to conventional sentiment analysis approaches due to its distinct characteristics. These characteristics can be categorised with respect to two aspects, *data* and *domain* as follows:

- *Data:* Tweets messages are very noisy in the sense that they often contain a large number of abbreviations, ill-formed words and irregular expressions. Moreover, tweets are usually written in informal English and composed of poorly-structured sentences. Such noisy characteristics often affects the performance of traditional sentiment analysis approaches.

- *Domain:* Twitter is an open social environment, where there are no restrictions on what users can tweet about. Approaches to analysing sentiment from a given Twitter corpus, are therefore required to adapt to the domain and topic of the tweets in that corpus. As discussed earlier, this is because the sentiment of words often changes with respect to their context in tweets.

The above challenges introduce the necessity of developing sentiment analysis approaches that are more *tolerant* to the noisy and sparse nature of tweets and more *flexible* to the dynamic and frequent emergence of new domains and topics on Twitter.

In this chapter, we reviewed existing approaches on sentiment analysis on Twitter, highlighting their strengths and weaknesses with respect the aforementioned challenges. We showed that these approaches can be categorised as (i) traditional (non-semantic) approaches (Section 2.2.1), and (ii) semantic approaches (Section 2.3).

Traditional approaches are those which rely on the syntactical parts in tweet texts that explicitly express sentiment (i.e., opinionated words, emoticons, intensifiers, etc.). Several machine learning and lexicon-based methods, which follow this principle, have been proposed.

Machine learning methods (Section 2.2.1.1) are usually supervised, and therefore, they require training from large annotated corpora. They are also domain-dependent, i.e, they require retraining for each new domain. Moreover, the data sparseness of tweets is a challenging problem to the supervised machine learning methods and its impact on the performance should be considered. However, the literature showed that very few works have tried to address this issue by devising sophisticated dimentionality techniques or

heavy feature engineering work (Section 2.2.1.1), whose applicability in the real word is yet to be investigated.

Lexicon-based methods (Section 2.2.1.2), on the other hand, rely on sentiment lexicons for sentiment assignment of the opinionated content in tweets. Therefore, these methods seem to be a better fit to Twitter than machine learning methods since they do not need training from labelled corpora. Nevertheless, we saw that the sentiment given by most existing lexicon-based methods is often too general and does not reflect the context of words in tweets. Also, sentiment lexicons are often composed of a fixed number of words that do not cover the wide range of new words and expressions that constantly emerge on Twitter. Hence, lexicon-based methods on Twitter should consider the context of words when detecting their sentiment in tweets rather than merely relying on the prior sentiment provided by general-purpose sentiment lexicons. In Section 2.2.1.2 we saw that a reasonable solution to the aforementioned limitation is by adapting sentiment lexicons to the domain or general context of the analysed tweets. However, only one of the reviewed methods considers updating its sentiment lexicon, and the updating method requires training from human-coded corpus [159].

In addition to the aforementioned limitations, traditional approaches (both machine learning and lexicon-based) *are fully dependent on the presence of words or syntactical features that explicitly reflect sentiment*. However, the sentiment of a word, as discussed earlier, is usually implicitly associated with its semantics in context. Words in general have different semantics in different contexts and, therefore different sentiment [33].

The above limitation has led to the emergent of various sentiment analysis models that extract the semantics of words based on their context and use it for sentiment analysis, creating a sub-filed in sentiment analysis conventionally called *semantic sentiment analysis* (Section 2.3).

Two types of approaches to semantic sentiment analysis have been described in this chapter, contextual and conceptual approaches.

Contextual semantic approaches (Section 2.3.1) derive the semantics of words from their co-occurrence patterns in text. They are based on the assumption that words that co-occur in similar contexts tend to have similar semantics. Extracting the semantics of words, therefore, often includes studying their co-occurrence patterns and consequently detecting their contextual sentiment. Three statistical techniques have been used in this line of work, Semantic Orientation from Association (SOA), Latent Semantic Analysis (LSA) and Latent Drichlit Allocation (LDA). SOA-based methods are domain- and

context-insensitive because they rely on the statistical correlations between words in very general and global data sources (e.g., the web). On the other hand, LSA and LDA based methods are corpus-based, that is, they extract the latent semantics of words form the studied corpus. Thus, they are usually better in capturing the sentiment of words at more specific context levels (i.e., corpus-level). Nevertheless, in addition to their computational runtime complexity, these methods usually neglect the syntactical structure of sentences and the orders of words in them. Therefore, they are likely to fail in detecting the sentiment in simple cases, such as the presence of negation.

Conceptual semantic approaches (Section 2.3.2), on the other hand, make use of external knowledge sources (ontologies and semantic networks) along with semantic reasoning techniques to extract the latent concepts in text with their associated sentiment [33, 55]. Unlike contextual semantic approaches, conceptual semantic approaches require a deep understanding of the syntactical structure of the sentence together with accurate detection and reasoning of the words' syntactical functions within it. On the one hand, such extensive analysis leads to a noticeable gain in the sentiment detection performance over the traditional and the contextual semantic approaches. It also allows for more fine-grained sentiment analysis tasks, such as sentence-, phrase- and entity-level sentiment analysis due to the ability of these approaches on detecting the context and semantics of words at these levels.

On the other hand, most conceptual semantic approaches are designed to work on formal text, where well-written and well-structured sentences are usually a prerequisite to proper functioning. Tweet messages, however, lack such formality and sentence structure as described earlier. Moreover, conceptual semantic approaches rely on heavy text processing, dimensionality reduction, and semantic parsing and reasoning techniques. Last but not least, these kind of approaches are limited by the external knowledge base used, which often provides a limited number of concepts or concept-associated sentiment. In Section 2.3.2, we have seen that the amount of work on conceptual sentiment analysis on Twitter is still little. Therefore, without a comprehensive evaluation, the performance in accuracy and time-efficiency of the conceptual semantic approaches on Twitter is yet to be measured.

Overall, based on our review in this chapter, a common and serious limitation of both, contextual and conceptual semantic approaches to sentiment analysis is that they are not tailored to Twitter and the like. Together with the limitations of traditional or non-

semantic approaches, these constitute the motivation behind the research work in this thesis.

2.4.1 Discussion

In this chapter we introduced an overview of existing works on sentiment analysis on Twitter, highlighting their properties and pointing to their strengths and weaknesses. Based on our review and Twitter's characteristics, one could conclude that, whether it is a machine learning or lexicon-based, a successful Twitter sentiment analysis approach should possess the following two main properties:

- Semantic sentiment analysis, a prerequisite for efficiently detecting the sentiment in tweets, whether it is associated with the context of words (contextual sentiment) or conveyed within the latent semantic concepts or semantic relations of words (conceptual sentiment).
- Tailored to Twitter, which denotes the ability of coping with the sparsity nature of tweets and language informality of their words, expression and sentences.

Table 1 categorises all types of approaches reviewed and analysed in this chapter with respect to the above properties.

	Traditional Approaches		**Semantic Approaches**	
Property	**Supervised-ML**	**Lexicon-based**	**Contextual**	**Conceptual**
Semantic Sentiment Analysis	No	No	Yes	Yes
Tailored to Twitter	No	Partly	No	No
Supervision	Supervised	Unsupervised	Unsupervised	Unsupervised
Domain Adaptation	Not Adaptable	Not Adaptable	Partly Adaptable	Adaptable
Runtime Complexity	Vary	Low	High	High

Table 1: Main and secondary properties of the four type of approaches to sentiment analysis on Twitter. ML: Machine Learning.

While the above two properties are useful to acquire since due to their impact on sentiment analysis performance, there are other secondary properties that sentiment analysis approaches on Twitter may or may not have, as shown in Table 1. This includes (i) the type of sentiment classifier (unsupervised, supervised), (ii) the ability to adapt to new domains or topics and (iii) the degree of complexity. These properties are secondary in the

sense that their absence in an approach may impact its performance positively or negatively based on the dataset analysed and the sentiment analysis application. For example, supervised machine learning methods are recommended over lexicon-based methods for polarity classification on a fixed and relatively small Twitter corpus of specific domain, where annotated tweet samples are provided. This is because the supervised methods can quickly adapt to the domain, topic, or context of the corpus, by training them from the annotated samples, and therefore, provides more accurate sentiment analysis than the lexicon-based methods. On the other hand, on a stream of tweets of diverse topics and domains, unsupervised or lexicon-based methods are more preferable than supervised approaches.

Based on Table 1 we can summarise the strengths and weaknesses of each approach reviewed in this chapter as follows:

1. *Traditional or non-semantic approaches* do not consider the semantics of words for sentiment analysis, but merely rely on the presence of words or syntactical features that explicitly reflect sentiment in tweets. These approaches can be divided into machine learning and lexicon-based approaches.

    a) Supervised machine learning approaches are domain-dependent, require training from annotated corpora and often face the data sparseness problem on Twitter.

    b) Lexicon-based approaches use general-purpose sentiment lexicons that most of the time, need to be extended and adapted to the context of words under analysis. Few existing works handle Twitter's noisy nature, and therefore, they tend to be more effective for Twitter than other approaches.

2. *Semantic sentiment analysis approaches* rely on the latent semantics of words that convey sentiment in text. They are generally designed to function on conventional text, which makes them less tailored to Twitter. These approaches can be divided into contextual and conceptual semantic approaches.

    c) Contextual semantic approaches are normally based on the bag-of-word paradigm, extract the contextual semantics from words' co-occurrence patterns, and often ignore cases like negation and mixed sentiment in tweets.

    d) Conceptual semantic approaches are based on the bag-of-concept paradigm, require external knowledge bases along with extensive text processing and semantic reasoning for semantic and sentiment extraction.

Lastly, the review and discussion conducted in this chapter suggest that sentiment analysis on Twitter still lacks the use of semantics (both, contextual and conceptual) for sentiment analysis on Twitter. Such lack of research works constitutes the main research problem and motivation behind the work in this thesis. It also forms the main research question of this thesis, as explained earlier in Chapter 1:

Could the semantics of words boost sentiment analysis performance on Twitter?

In the following chapters we introduce new approaches and solutions to addressing the above question, along with experimental work and evaluation conducted to measure the efficiency of our proposed approaches.

Part II

SEMANTIC SENTIMENT ANALYSIS OF TWITTER

All our work, our whole life is a matter of semantics, because words are the tools with which we work, the material out of which laws are made, out of which the Constitution was written. Everything depends on our understanding of them.

Felix Frankfurter

3

CONTEXTUAL SEMANTICS FOR SENTIMENT ANALYSIS OF TWITTER

In this chapter we explore the use of contextual semantics in lexicon-based sentiment analysis on Twitter. In essence, we propose a new approach for capturing the contextual semantics and sentiment of words from tweets, and evaluate the effectiveness of the proposed approach in three sentiment analysis tasks on Twitter. Our conclusion is that lexicon-based methods that considers contextual semantics produce, in most cases, a higher performance than those that merely rely on affect words for sentiment analysis.

3.1 INTRODUCTION

Traditional Lexicon-based approaches to sentiment analysis on Twitter have two main limitations as discussed in Chapter 2. Firstly, the number of words in the sentiment lexicons is finite. This may constitute a problem when extracting sentiment from very dynamic environments such as Twitter, where new terms, abbreviations and malformed words constantly emerge. Secondly and more importantly, lexicon-based approaches assign static sentiment values to words, regardless of the contexts in which these words are used and the semantics they convey. In many cases however, the sentiment of a word is implicitly associated with its semantics in a given context (aka, contextual semantics) [167, 33, 139]. For example, the word "`great`" should be negative in the context of a "`problem`", and positive in the context of a "`smile`".

In this chapter, we investigate extracting and using the contextual semantics of words in sentiment analysis on Twitter, aiming mainly at addressing the above limitations of traditional lexicon-based approaches and consequently improving their performances.

The research question we aim to address in this chapter is:

RQ1 ***Could the contextual semantics of words enhance lexicon-based sentiment analysis performance?***

To this end, we propose a semantic representation model called *SentiCircle*, which captures the contextual semantics of words from their co-occurrence patterns in tweets and calculates their sentiment orientation accordingly. Remember that the main principle behind the notion of contextual semantics comes from the dictum-"You shall know a word by the company it keeps!" [52]. This suggests that words that co-occur in a given context tend to have a certain relation or semantic influence [177, 167], which we try to capture with the SentiCircle model.

To assess whether the use of contextual semantics helps enhance the performance of lexicon-based approaches we evaluate the SentiCircle model in three sentiment analysis tasks on Twitter. Entity-level sentiment analysis (detect the sentiment of individual named-entities on Twitter), tweet-level sentiment analysis (i.e., detect the overall sentiment of a tweet) and context-aware sentiment lexicon adaptation (i.e., amend the prior sentiment of words in a given sentiment lexicon with respect to their contextual semantics in the dataset under analysis). To this end, we design several lexicon-based and rule-based methods on top of SentiCirlces for each sentiment analysis tasks.

Evaluation under each sentiment analysis task includes using multiple Twitter datasets and sentiment lexicons.

Results show that our proposed methods, based on SentiCircles, overcome the above two limitations that non-semantic lexicon-based approaches often face on Twitter. First, for entity-level sentiment analysis our methods outperform all other baselines by 30-40% for subjectivity detection, and by 2-15% for polarity detection in average F-measure. Secondly, for tweet-level sentiment analysis results showed that SentiCircle outperforms the state-of-the-art, SentiStrength, in accuracy by 4-5% in two datasets, and in F-measure by 1% in one dataset. Lastly, sentiment lexicons adapted by our proposed methods, when used for polarity detection, improve accuracy by 1-2%, and F-measure by 1-4% in two datasets over using general lexicons with no semantic adaptation.

The rest of this chapter is organised as follows: Section 3.2 presents our proposed SentiCircle representation and its use for capturing the contextual sentiment of words. Section 3.3.2 demonstrates how to use SentiCircles for tweet- and entity-level sentiment analysis. Evaluation setup and results are covered in Sections 3.3.3 and 3.3.4. Adapting sentiment lexicons with SentiCircles is presented along with evaluation results and finding in Section 3.4. Runtime analysis of SentiCircles is introduced in Section 3.5. Our findings are discussed in Section 3.6. Finally, we summarise our work in this chapter in Section 3.7.

3.2 SENTICIRCLES: CAPTURING AND REPRESENTING CONTEXTUAL SEMANTICS FOR SENTIMENT ANALYSIS

In this section we introduce our SentiCircle representation and its use for capturing the contextual semantics and sentiment of words.

3.2.1 Overview

SentiCircle aims to represent the sentiment orientation of words with respect to their contextual semantics. The main notion behind this is that the sentiment of a term is not static, as in traditional lexicon-based approaches, but rather depends on the context in which the term is used, i.e., it depends on its contextual semantics. For example, most existing sentiment analysis methods fail to detect the sentiment of the tweet "`#Syria. Execution continues with smile! :( #ISIS`", since they consider the existence of the word "`smile`" positive, even though it is used within the context of the negative word "`Execution`".

To capture the contextual semantics of a term, we follow the distributional semantic hypothesis that words that occur in similar contexts tend to have similar meaning [177, 167]. Therefore, the contextual semantics of a term in our approach is computed from its co-occurrence patterns with other terms. Note that we define context as a textual corpus or a set of tweets.

Thus, in order to understand the semantics and the sentiment of a target word like "`ISIS`"(Islamic State in Iraq and Syria), our method relies on the words that co-occur with the target word in a given collection of tweets. These co-occurrences are mathematically represented as a 2D geometric circle; where the target word ("`ISIS`") is at the centre of the circle and each point in the circle represents a context word that co-occur with "`ISIS`" in the tweet collection (see Figure 9). The position of each context term (e.g., "`Kill`") in the circle determines its importance and sentiment toward the target term. Such a position can be defined by an angle (representing the prior sentiment of the context term) and a radius (representing the correlation between the context term and the target term), as will be explained in the subsequent section.

The rationale behind using this circular representation shape, which will become clearer later, is to benefit from the trigonometric properties it offers for estimating the sentiment orientation, and strength, of terms. It also enables us to calculate the impact of context

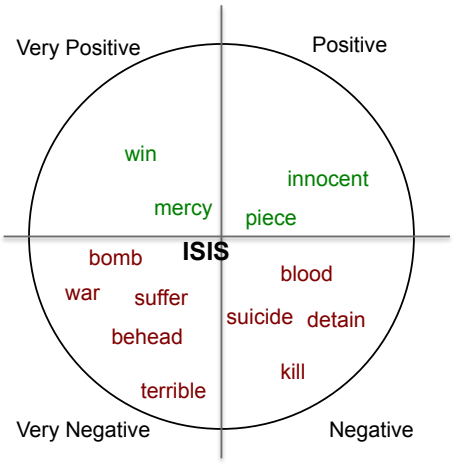


Figure 9: Example SentiCircle for the word "ISIS". Terms positioned in the upper half of the circle have positive sentiment while terms in lower half have negative sentiment.

words on the sentiment orientation (i.e., positive, negative, neutral) and on the sentiment strength (e.g., weak positive, strong negative, etc) of a target-word separately, which is difficult to do with traditional vector representations.

In the rest of this section we describe our proposed pipeline for constructing the SentiCircle of a given term and how to use it to measure the term's contextual-sentiment orientation and strength.

3.2.2 SentiCircle Construction Pipeline

Figure 10 shows the pipeline of extracting and using SentiCircles to calculate the contextual semantics and sentiment of terms, which can be summarised in the following steps:

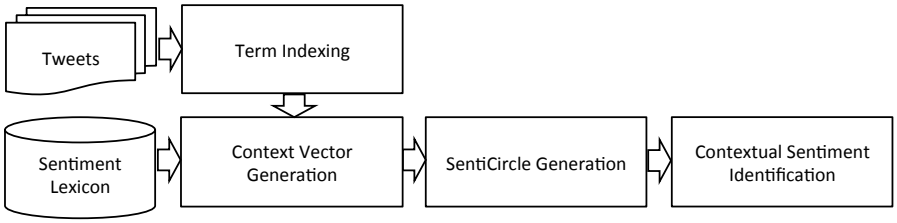


Figure 10: The systematic pipeline of the SentiCircle approach for contextual sentiment analysis.

- Term Indexing: This step creates an index of terms (term-index) from a collection of tweet messages. Several text processing procedures are applied during the pro-

cess such as: Filtering Non-English terms,[1] Part-Of-Speech tagging and Negation (Section 3.2.2.1).

- Context Vector Generation: This step represents each term m as a vector of all its context terms (i.e., terms that occur with m in the same context) in the tweets. For each context term, we calculate (i) its degree of correlation to the term m and (ii) a prior (initial) sentiment score using an external sentiment lexicon. (Section 3.2.2.2).

- SentiCircle Generation: This step converts the context vector of m into a 2D geometric circle, which is composed of points denoting the context terms of m. Each context term is located in the circle based on its angle (defined by its prior sentiment), and its radius (defined by its degree of correlation with the term m, which is measured as explained below) (Section 3.2.2.3).

- Contextual Sentiment Identification: Here, we calculate the contextual sentiment of the term m using the trigonometric properties of its SentiCircle. (Section 3.2.2.4).

Word Context: As mentioned before, the SentiCircle model aims to capture the sentiment of a word based on its context in the tweet data. The context of a word can be usually defined at different levels, including: (i) *The sentence-level*: here the word's context is extracted from the sentence in the tweet where the word occurs. (ii) *The tweet-level*: here the context of the word is extracted from the tweet where the word occurs (given that a tweet may consist of multiple sentences and/or phrases). (iii) *The corpus-level*: the context here is extracted from a collection of tweets in which the word occur.

In this work we choose to extract the context of a word in the SentiCircle model at the corpus-level, as will be explained in Section 3.2.2.2. This is because our aim here is to extract the collective or the aggregated sentiment of the word in a given Twitter corpus.

3.2.2.1 *Term Indexing*

The first step in our pipeline is to index the unique terms given collection of raw tweets. To this end, we apply the following processing steps:

- Remove all non-ASCII and non-English characters.

- Extract the part-of-speech tag (POS tag) of all terms. To this end we use the TweetNLP POS tagger,[2] which is built to work specifically on tweets data.

1 By removing non-ASCII characters.

2 http://www.ark.cs.cmu.edu/TweetNLP/

- Find and index terms, which appear in the tweet within the vicinity of a negation, as *negated terms*. The purpose of this step, as will be shown shortly, is to inverse the sentiment of such terms when constructing SentiCircles. For example, in the tweet "`iPad is not amazing!`", the term "`amazing`" is preceded by a negation. Therefore, we inverse its original sentiment from positive to negative. The negation words (e.g., not, don't, can't, etc.) are collected from the General Inquirer under the NOTLW category.[3]

  The output of this step is an index of unique and processed terms which we refer to as the *Term-Index*.

3.2.2.2 *Context Vector Generation*

The SentiCircle representation is based on co-occurrences between terms in tweets and the influence these terms have on each others. Therefore, this step aims to represent each term as a vector of its co-occurrences with all other terms in the tweet collection. We refer to this vector as *Context Vector* and it can be defined and constructed as below.

Definition (Context Vector) Given a set of tweet messages $\mathcal{T}$ and Term-Index $\mathcal{D}$, the context vector of a term $m \in \mathcal{D}$ is a vector $\mathbf{c} = (c_1, c_2, ..., c_n)$ of terms that occur with m in any tweet in $\mathcal{T}$.

The contextual semantics of m is determined by its semantic relation to each context term $c_i \in \mathbf{c}$. We compute the individual semantic relation between m and a context term c_i, by assigning the following two main features to c_i:

- Prior Sentiment Score: Each context term c_i is assigned a prior sentiment score based on its POS tag(s) by using one of the three external sentiment lexicons used in this work (Section 4.2.3.3). If this term c_i appears in the vicinity of a negation, its prior sentiment score is negated

- Term Degree of Correlation (*TDOC*): This feature represents the degree of correlation between a term m and its context term $c_i \in \mathbf{c}$ (i.e., how important c_i is to m). Inspired by the TF-IDF weighting scheme, we compute the value of this feature as:

$$\mathrm{TDOC}(m, c_i) = f(c_i, m) \times \log \frac{N}{N_{c_i}} \tag{3}$$

  where $f(c_i, m)$ is the number of times c_i occurs with m in tweets, N is the total number of terms, and N_{c_i} is the total number of terms that occur with c_i.

3 http://www.wjh.harvard.edu/~inquirer/NotLw.html

3.2.2.3 *SentiCircle Generation*

Now we have for each term m a vector of its context terms $\mathbf{c}$ along with the two semantic mutual features between m and each $c_i \in \mathbf{c}$. From this information, we represent the contextual semantics of the term m as a geometric circle; *SentiCircle*, where the term is situated in the centre of the circle, and each point around it represents a context term c_i. The position of c_i is defined jointly by its prior sentiment and its term degree of correlation (TDOC).

Formally, a SentiCircle in a polar coordinate system can be represented with the following equation:

$$r^2 - 2rr_0\cos(\theta - \phi) + r_0^2 = a^2 \tag{4}$$

Where a is the radius of the circle, (r_0, ϕ) is the polar coordinate of the center of the circle, and (r, θ) is the polar coordinate of a context term on the circle. For simplicity, we assume that our SentiCircles are centred at the origin (i.e, $r_0 = 0$).

Hence, to build a SentiCircle for a term m, we only need to calculate, for each context term c_i in the a radius r_i and an angle θ_i. To do that, we use the prior sentiment score and the T-DOC value of the term c_i as:

$$\begin{aligned} r_i &= \text{TDOC}(m, c_i) \\ \theta_i &= \text{Prior\_Sentiment}(c_i) * \pi \end{aligned} \tag{5}$$

We normalise the radii of all the terms in a SentiCircle to a scale between 0 and 1. Hence, the radius a of any SentiCircle is equal to 1. Also, all angle values are in radians.

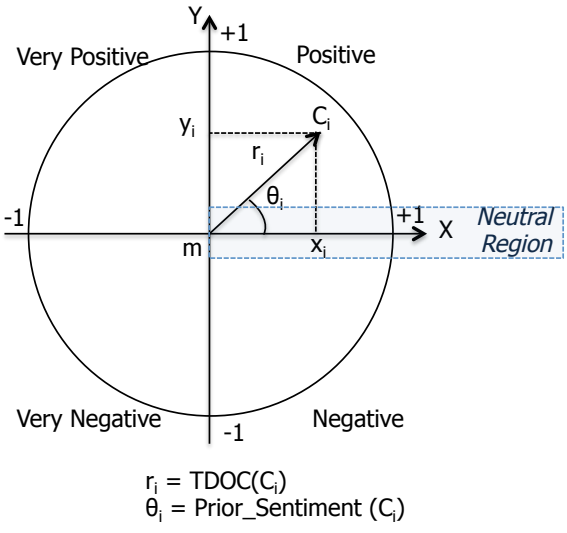


Figure 11: SentiCircle of a term m.

The SentiCircle in the *polar coordinate system* can be divided into four sentiment quadrants as shown in Figure 11. Terms in the two upper quadrants have a positive sentiment ($\sin\theta > 0$), with upper left quadrant representing stronger positive sentiment since it has larger angle values than those in the top right quadrant. Similarly, terms in the two lower quadrants have negative sentiment values ($\sin\theta < 0$). Although the radius of the SentiCircle of any term m equals to 1, points representing context terms of m in the circle have different radii ($0 \leqslant r_i \leqslant 1$), which reflect how important a context term is to m. The larger the radius, the more important the context term to m.

We can move from the *polar coordinate system* to the *Cartesian coordinate system* by simply using the trigonometric functions `sine` and `cosine` as:

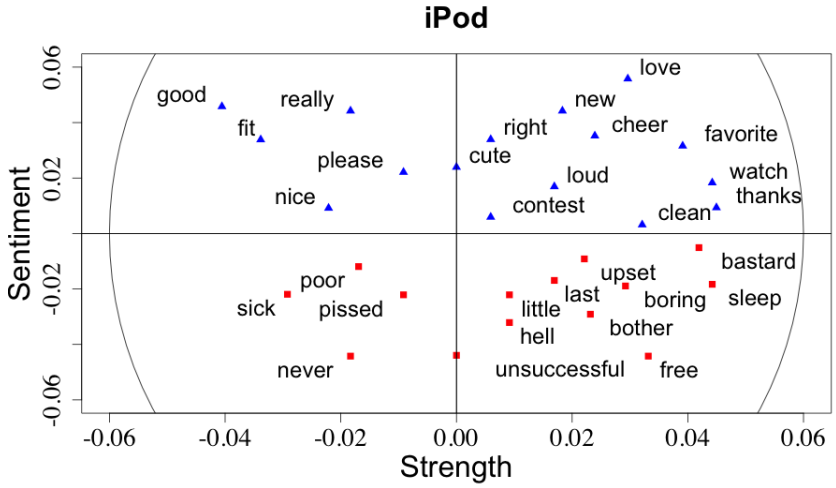
$$x_i = r_i \cos\theta_i \qquad\qquad y_i = r_i \sin\theta_i \tag{6}$$

Moving to the *Cartesian coordinate system* allows us to use the trigonometric properties of the circle to encode the contextual semantics of a term in the circle as sentiment orientation and sentiment strength. Y-axis in the Cartesian coordinate system defines the sentiment of the term, i.e., a positive y value denotes a positive sentiment and vice versa. The X-axis defines the sentiment strength of the term. The smaller the x value, the stronger the sentiment.[4] Moreover, a small region called the "*Neutral Region*" can be defined. This region, as shown in Figure 11, is located very close to X-axis in the "*Positive*" and the "*Negative*" quadrants only, where terms lie in this region have very weak sentiment (i.e., $|\theta| \approx 0$). The "*Neutral Region*" has a crucial role in measuring the overall sentiment of a given SentiCircle as will be shown in the subsequent sections.

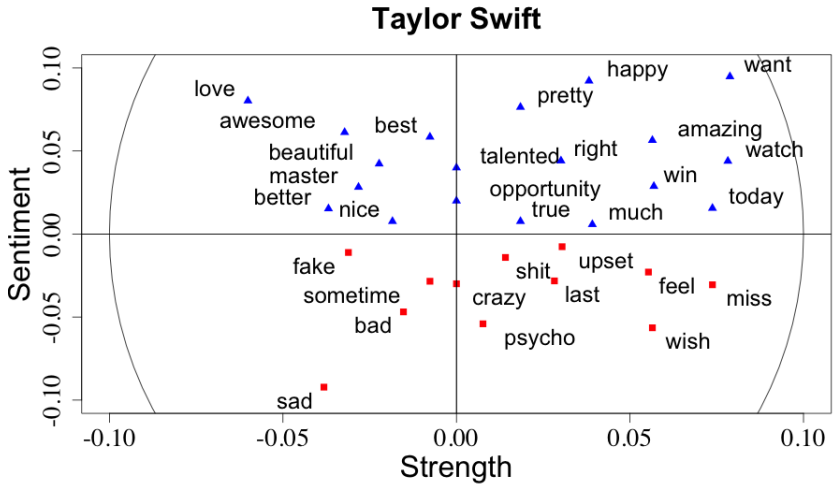
Note that in the extreme case, where $r_i = 1$ and $\theta_i = \pi$ we position the context term c_i in the "*Very Positive*" or the "*Very Negative*" quadrants based on the sign of its prior sentiment score.

Figure 12 shows the SentiCircles of the entities "`iPod`" and "`Taylor Swift`". Terms (i.e., points) inside each circle are positioned in a way that represents their sentiment scores and their importance (degree of correlation) to the entity. For example, "`Awesome`" in the SentiCircle of "`Taylor Swift`" has a positive sentiment and a high importance score, hence it is positioned in the "*Very Positive*" quadrant (see Figure 12(b)). The word "`Pretty`", in the same circle, also has positive sentiment, but it has lower importance score than the word "`Awesome`", hence it is positioned in the "*Positive*" quadrant. We also notice that there are some words that appear in both circles, but in different positions. For example, the word "`Love`" has a stronger positive sentiment strength with "`Taylor`

4 This is because $\cos\theta < 0$ for large angles.



(a) iPod.



(b) Taylor Swift.

Figure 12: Example SentiCircles for "iPod" and "Taylor Swift". We have removed points near the origin for easy visualisation. Dots in the upper half of the circle (triangles) represent terms bearing a positive sentiment while dots in the lower half (squares) are terms bearing a negative sentiment.

`Swift`" compared to "`iPod`", although it has a positive sentiment (similar y-value) in both circles.

As described earlier, the contribution of both quantities (prior sentiment and degree of correlation) is calculated and represented in the SentiCircle separately by means of the projection of the context term along X-axe (sentiment strength) and Y-axe (sentiment orientation). Such level of granularity is crucial when we need, for example, to filter those context words that have low contribution towards the sentiment orientations or strength of the target word.

3.2.2.4 *Senti-Median: The Overall Contextual Sentiment Value*

The previous examples in Section 3.2.2.3 show that, although we use external lexicons to assign initial sentiment scores to terms, our SentiCircle representation is able to amend these scores according to the context in which each term is used. The last step of our pipeline is to compute the overall contextual sentiment of the term based on its SentiCircle. To this end, we use the *Senti-Median* metric.

Senti-Median We now have the SentiCircle of a term m which is composed by the set of (x, y) Cartesian coordinates of all the context terms of m, where the y value represents the sentiment and the x value represents the sentiment strength. An effective way to approximate the overall sentiment of a given SentiCircle is by calculating the geometric median of all its points. Formally, for a given set of n points $(p_1, p_2, ..., p_n)$ in a SentiCircle Ω, the 2D geometric median g is defined as:

$$g = \arg\min_{g \in \mathbb{R}^2} \sum_{i=1}^{n} \|p_i - g\|_2, \tag{7}$$

where the geometric median is a point $g = (x_k, y_k)$ in which its Euclidean distances to all the points p_i is minimum. We call the geometric median g the **Senti-Median** as it captures the sentiment (y-coordinate) and the sentiment strength (x-coordinate) of the SentiCircle of a given term m.

Following the representation provided in Figure 11, the sentiment of the term m is dependent on whether the Senti-Median g lies inside the neutral region, the positive quadrants, or the negative quadrants. Formally, given a Senti-Median g_m of a term m, the term-sentiment function $\mathcal{L}$ works as:

$$\mathcal{L}(g_m) = \begin{cases} \text{negative} & \text{if } y_g < -\lambda \\ \text{positive} & \text{if } y_g > +\lambda \\ \text{neutral} & \text{if } |y_g| \leqslant \lambda \;\&\; x_g \geqslant 0 \end{cases} \tag{8}$$

where λ is the threshold that defines the Y-axis boundary of the neutral region. Section 3.3.3.4 illustrates how this threshold is computed.

3.3 SENTICIRCLES FOR SENTIMENT ANALYSIS

So far in this Chapter, we introduced our SentiCircle approach and described how it can be used to capture the contextual semantics and sentiment of words in Twitter. In this section we show how to use SentiCircles in two different sentiment analysis tasks; entity- and tweet-level sentiment detection.

3.3.1 Entity-level Sentiment Detection

In this section we explain how to use the SentiCircle representation for sentiment analysis at the entity level, i.e., identifying the sentiment of individual named-entities in a given tweet collection.

Given an entity, $e_i \in \mathcal{E}$, and its corresponding SentiCircle representation, the sentiment of the entity is given by the position the Senti-Median g of the entity's SentiCircle (see Function 8). If the Senti-Median g lies inside the "*Neutral Region*", the entity will have a **neutral sentiment**. If g lies in one of the positive quadrants, the entity will have a **positive sentiment** and, if g lies in the negative quadrants, the entity will have a **negative sentiment**.

3.3.2 Tweet-level Sentiment Detection

There are several ways in which the SentiCircle representations of the terms that compose the tweet can be used to determine the tweet's overall sentiment. For example, the tweet "`iPhone and iPad are amazing`" contains five terms. Each of these terms has an associated SentiCircle representation. These five SentiCircles can be combined in different ways in order to extract the sentiment associated to the tweet. In this section we propose

three different methods that use the SentiCircle representation for tweet-level sentiment detection.

3.3.2.1 *The Median Method*

This method works by representing each tweet message $t_i \in \mathcal{T}$ as a vector of Senti-Medians $\mathbf{g} = (g_1, g_2, ..., g_n)$ of size n, where n is the number of terms that compose the tweet and g_j is the Senti-Median of the SentiCircle associated to term m_j. Equation 7 is then used to calculate the median point q of $\mathbf{g}$, which we use to determine the overall sentiment of tweet t_i using Function 8.

3.3.2.2 *The Pivot Method*

This method favours some terms in a tweet over others, based on the assumption that sentiment is often expressed towards one or more specific targets, which we refer to as "`Pivot`" terms. In the tweet example above, there are two pivot terms, "`iPhone`" and "`iPad`" since the sentiment word "`amazing`" is used to describe both of them. Hence, the method works by (1) extracting all pivot terms in a tweet and; (2) accumulating, for each sentiment label, the sentiment impact that each pivot term receives from other terms. The overall sentiment of a tweet corresponds to the sentiment label with the highest sentiment impact.

Opinion target identification is a challenging task and is beyond the scope of our work in this thesis. For simplicity, we assume that the pivot terms are those having the POS tags: {*Common Noun, Proper Noun, Pronoun*} in a tweet. For each candidate pivot term, we build a SentiCircle from which the sentiment impact that a pivot term receives from all the other terms in a tweet can be computed. Formally, the Pivot-Method seeks to find the sentiment $\hat{s}$ that receives the maximum sentiment impact within a tweet as:

$$\begin{aligned} \hat{s} &= \arg\max_{s \in \mathcal{S}} \mathcal{H}_s(\mathbf{p}) \\ &= \arg\max_{s \in \mathcal{S}} \sum_{i}^{N_p} \sum_{j}^{N_w} \mathcal{H}_s(p_i, w_j), \end{aligned} \tag{9}$$

where $s \in \mathcal{S} = \{\mathtt{Positive}, \mathtt{Negative}, \mathtt{Neutral}\}$ is the sentiment label, $\mathbf{p}$ is a vector of all pivot terms in a tweet, $N_{\mathbf{p}}$ and N_w are the sets of the pivot terms and the remaining terms in a tweet respectively. $\mathcal{H}_s(p_i, w_j)$ is the sentiment impact function, which returns the sentiment impact of a term w_j in the SentiCircle of a pivot term p_i. The sentiment impact of a term within a SentiCircle of a pivot term is the term's Euclidean distance

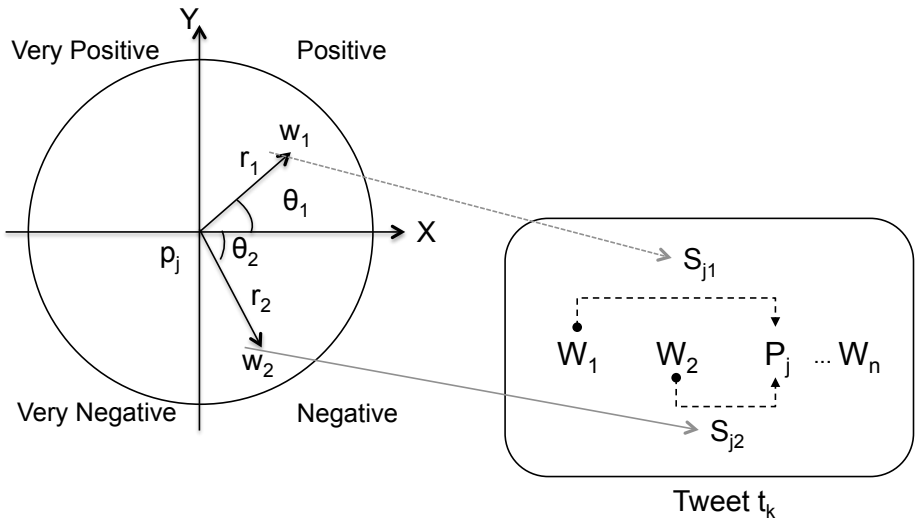


Figure 13: The Pivot Method. The figure shows a SentiCircle of a pivot term p_j. The sentiment strength S_{j1} of word w_1 with respective to the pivot term p_j is the radius r_1 in the SentiCircle, and likewise for S_{j2}.

from the origin (i.e., the term's radius) as shown in Figure 13. Note that the impact value is doubled for all terms located either in the "*Very Positive*" or in the "*Very Negative*" quadrants.

3.3.2.3 *The Pivot-Hybrid Method*

The Pivot Method, as described in the previous section, relies on both, the syntactic structure of a tweet and the sentiment relations among its terms. As such, it may suffer from the lack of pivot terms when the tweet message is too short or it contains many ill-formed words. In such a case, we resort to the Median method, and call this the `Pivot-Hybrid` method.

3.3.3 Evaluation Setup

As mentioned in Section 3.3, the contextual semantics captured by the SentiCircle representation are based on terms co-occurrence from the corpus and an initial set of sentiment weights from a sentiment lexicon. We propose an evaluation set up that uses three different corpora (collections of tweets) and three different generic sentiment lexicons. This enables us to assess the influence of different corpora and lexicons on the performance of our SentiCircle approach.

3.3.3.1 *Datasets*

In this section, we present the three datasets used for the evaluation; OMD, HCR and STS-Gold. We use the OMD [47] and HCR [152] datasets[5] to assess the performance of our approach at the tweet level only since they provide human annotations for tweets but not for entities (i.e., each tweet is assigned a positive, negative or neutral sentiment label).[6]

Due to the lack of gold-standard datasets for evaluating entity-level sentiment analysis models, we have generated an additional dataset (STS-Gold) [138].[7] This dataset contains both, tweet and entity sentiment ratings and therefore, we use it to assess the performance of SentiCircles at both the entity and the tweet levels.

Numbers of positive and negative tweets within the three datasets are summarised in Table 2 and further described below:

Dataset	Tweets	Positive	Negative
OMD [47]	1081	393	688
HCR [152]	1354	397	957
STS-Gold [138]	2034	632	1402

Table 2: Twitter datasets used for the evaluation.

Obama-McCain Debate Dataset (OMD)

This dataset was constructed from 3,238 tweets crawled during the first U.S. presidential TV debate in September 2008 [47]. Sentiment ratings of these tweets were acquired using Amazon Mechanical Turk, where each tweet was rated by one or more voter as either positive, negative, mixed, or other. We only keep those tweets rated by at least three voters with two-third of the votes being either positive or negative to ensure their sentiment polarity. This resulted in a set of 1,081 tweets with 393 positive and 688 negative ones.

Health Care Reform Dataset (HCR)

HCR dataset was built by crawling tweets containing the hashtag "#hcr" (health care reform) in March 2010 [14]. A subset of this corpus was manually annotated with three polarity labels (positive, negative, neutral) and split into training and test sets. In this

5 https://bitbucket.org/speriosu/updown

6 We refer the reader to Appendix A for details about the construction and the annotation of the HCR and OMD datasets.

7 http://tweenator.com

work we focus on identifying positive and negative tweets and therefore we exclude neutral tweets from this dataset. This resulted in a set of 1354 tweets, 397 positive and 957 negative.

Standford Sentiment Gold Standard Dataset (STS-Gold)

We constructed this dataset as a subset of the the Stanford Twitter Sentiment Corpus (STS) [60]. It contains 2,034 tweets (632 positive and 1402 negative) and 58 entities (13 negative, 27 positive and 18 neutral entities) manually annotated by three different human evaluators. To avoid noisy or misleading data in the created dataset, the entities and tweets selected for these dataset are those for which the three human evaluators agreed on the same sentiment label. To our knowledge, the STS-Gold dataset is the first publicly-available evaluation dataset that contains both, tweet and entity sentiment ratings.

In the following we describe the construction and the annotation processes of the STS-Gold.

1. Data Acquisition

To construct this dataset, we first extracted all named entities from a collection of 180K tweets randomly selected from the original Stanford Twitter corpus. To this end, we used AlchemyAPI,[8] an online service that allows for the extraction of entities from text along with their associated semantic concept class (e.g., Person, Company, City). After that, we identified the top most frequent semantic concepts and, selected under each of them, the top 2 most frequent and 2 mid-frequent entities. For example, for the semantic concept *Person* we selected the top most frequent entities (Taylor Swift and Obama) as well as two mid frequent entities (Oprah and Lebron). This resulted in 28 different entities along with their 7 associated concepts as shown in Table 3.

The next step was to construct and prepare a collection of tweets for sentiment annotation, ensuring that each tweet in the collection contains one or more of the 28 entities listed in Table 3. To this aim, we randomly selected 100 tweets from the remaining part of the STS corpus for each of the 28 entities, i.e., a total of 2,800 tweets. We further added another 200 tweets without specific reference to any entities to add up a total of 3,000 tweets. Afterwards, we applied AlchemyAPI on the selected 3,000 tweets. Apart from the initial 28 entities the extraction tool returned 119 additional entities, providing a total of

8 www.alchemyapi.com

Concept	Top 2 Entities	Mid 2 Entities
Person	Taylor Swift, Obama	Oprah, Lebron
Company	Facebook, Youtube	Starbucks, McDonalds
City	London, Vegas	Sydney, Seattle
Country	England, US	Brazil, Scotland
Organisation	Lakers, Cavs	Nasa, UN
Technology	iPhone, iPod	Xbox, PSP
HealthCondition	Headache, Flu	Cancer, Fever

Table 3: 28 Entities, with their semantic concepts, used to build STS-Gold.

147 entities for the 3,000 selected tweets.

2. Data Annotation

We asked three graduate students to manually label each of the 3,000 tweets with one of the five classes: (`Negative, Positive, Neutral, Mixed` and `Other`). The "`Mixed`" label was assigned to tweets containing mixed sentiment and "`Other`" to those that were difficult to decide on a proper label. The students were also asked to annotate each entity contained in a tweet with the same five sentiment classes. The students were provided with a booklet explaining both the tweet-level and the entity-level annotation tasks. The booklet also contains a list of key instructions as shown in Appendix B. It is worth noting that the annotation was done using Tweenator,[9] an online tool that we previously built to annotate tweet messages [137].

We measured the inter-annotation agreement using the Krippendorff's alpha metric [85], obtaining an agreement of $\alpha_t = 0.765$ for the tweet-level annotation task. For the entity-level annotation task, if we measured sentiment of entity for each individual tweet, we only obtained $\alpha_e = 0.416$ which is relatively low for the annotated data to be used. However, if we measured the aggregated sentiment for each entity, we got a very high inter-annotator agreement of $\alpha_e = 0.964$.

To construct the final STS-Gold dataset we selected those tweets and entities for which our three annotators agreed on the sentiment labels, discarding any possible noisy data from the constructed dataset. As shown in Table 4 the STS-Gold dataset contains 13

9 http://tweenator.com

negative, 27 positive and 18 neutral entities as well as 1,402 negative, 632 positive and 77 neutral tweets. The STS-Gold dataset contains independent sentiment labels for tweets and entities, supporting the evaluation of tweet-based as well as entity-based Twitter sentiment analysis models.

Class	Negative	Positive	Neutral	Mixed	Other
No. of Entities	13	27	18	-	-
No. of Tweets	1402	632	77	90	4

Table 4: Number of tweets and entities under each class in the STS-Gold dataset.

3.3.3.2 *Sentiment Lexicons*

As describe in Section 3.2.2.3, initial sentiments of terms in SentiCircle are extracted from a sentiment lexicon (prior sentiment). We evaluate our approach using three external sentiment lexicons in order to study how the different prior sentiment scores of terms influence the performance of the SentiCircle representation for sentiment analysis. The aim is to investigate the ability of SentiCircles in updating these *context-free* prior sentiment scores based on the contextual semantics extracted from different tweets corpora. We selected three state-of-art lexicons for this study: (i) the SentiWordNet lexicon [8], (ii) the MPQA subjectivity lexicon [176] and, (iii) Thelwall-Lexicon [159, 160].

3.3.3.3 *Baselines*

We compare the performance of our proposed SentiCircle representation when being used for tweet and entity sentiment analysis against the following baselines:

Lexicon Labelling Method: This method uses the MPQA and the SentiWordNet lexicons to extract the sentiment of a given text. If a tweet contains more positive words than negative ones, it is labelled as positive, and vice versa. For entity-level sentiment detection, the sentiment label of an entity is assigned based on the number of positive and negative words that co-occur with the entity in its associated tweets. In our evaluation, we refer to the method that uses the MPQA lexicon as *MPQA-Method* and to the method that uses the SentiWordNet lexicon as *SentiWordNet-Method*.

SentiStrength: SentiStength [159, 160] is a state-of-the-art lexicon-based sentiment detection approach. It assigns to each tweet two sentiment strengths: a negative strength between -1 (not negative) to -5 (extremely negative) and a positive strength between +1

(not positive) to +5 (extremely positive). To use SentiStrength for tweet-level sentiment detection, a tweet is considered positive if its positive sentiment strength is 1.5 times higher than the negative one, and negative otherwise[10]. For entity-level sentiment detection, the sentiment of an entity is assigned based on the total number of positive, negative tweets in which the entity occurs. The entity is positive if it occurs with more positive tweets than negative ones, and vice versa. The entity is neutral if it occurs in a similar number of positive and negative tweets. It is worth noticing that SentiStrength requires the manually-defined lexical rules, such as the existence of emoticons, intensifiers, negation and booster words (e.g, absolutely, extremely), to compute the average sentiment strength of a tweet.

3.3.3.4 *Thresholds and Parameters Tuning*

When computing the sentiment of a point within a SentiCircle (Function 8) it is necessary to determine beforehand the geometric boundaries of the neutral region (the region defining the neutral terms) within the SentiCircle. While the boundaries of the neutral region are fixed for the X-axis $[0, 1]$ (see Section 3.2.2.3), the boundaries of the Y-axis need to be determined. We have observed that the neutral area of a SentiCircle is defined by a high density of terms, since the number of neutral terms in the SentiCircle is usually larger than the number of positive and negative terms.

The limits of the neutral region vary from one SentiCircle to another. For simplicity, we assume the same neutral region boundary for all SentiCircles emerging from the same corpus and sentiment lexicon. To compute these thresholds we first build the SentiCircle of the complete corpus by merging all SentiCircles of each individual term and then we plot the density distribution of the terms within the constructed SentiCircle. The boundaries of the neutral area delimited by an increase/decrease in the density of terms.

	SentiWordNet	**MPQA**	**Thelwall-Lexicon**
OMD	[-0.01, 0.01]	[-0.01, 0.01]	[-0.01, 0.01]
HCR	[-0.1, 0.1]	[-0.05, 0.05]	[-0.05, 0.05]
STS-Gold	[-0.1, 0.1]	[-0.05, 0.05]	[-0.001, 0.001]

Table 5: Neutral region boundaries for Y-axis.

10 http://sentistrength.wlv.ac.uk/documentation/SentiStrengthJavaManual.doc

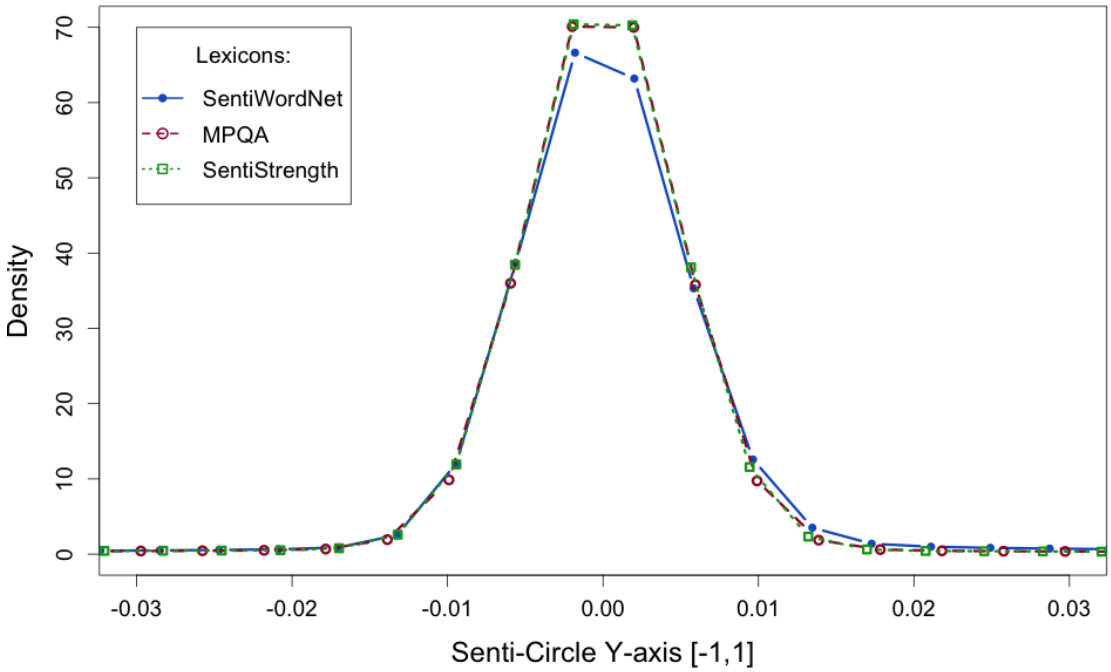


Figure 14: Density geometric distribution of terms on the OMD dataset.

Figure 21 shows the three density distribution plots for the OMD dataset with SentiWordNet, MPQA and Thelwall lexicons. The boundaries of the neutral area are delimited by the density increase, falling in the $[-0.01, 0.01]$ range. Note that the generated SentiCircles vary depending on the corpus and sentiment lexicon. For evaluation, we have computed nine different neutral regions, one for each corpus and sentiment lexicon used as shown in Table 5.

3.3.4 Evaluation Results

We report the performance of our proposed approaches in comparison with the baselines in both the entity- and tweet-level sentiment detection tasks. For entity-level sentiment detection, we conduct experiments on the STS-Gold dataset, while for tweet-level sentiment detection, we use the OMD, HCR and STS-Gold datasets.

3.3.4.1 *Entity-Level Sentiment Detection*

For entity-level sentiment detection, we employ our proposed Senti-Median method (see Section 3.2.2.4 and Section 3.3.1) with SentiWordNet, MPQA and Thelwall lexicons to

identify the overall sentiment of the SentiCircle of a given entity. We report the results in accuracy, precision, recall and F-measure on two identification tasks: **subjectivity detection**, which identifies whether a given entity is subjective (positive or negative) or objective (neutral). The second task is **polarity detection**, which identifies whether the entity has positive or negative sentiment. Both identification tasks are applied on 58 different entities (see Section 3.3.3.1).

Subjectivity Classification (Subjective vs. Objective)

Methods	Accuracy	Subjective			Objective			Average		
		P	R	F1	P	R	F1	P	R	F1
MPQA-Method	63.79	67.27	92.50	77.89	0	0	0	33.64	46.25	38.95
SentiWordNet-Method	63.79	67.27	92.50	77.89	0	0	0	33.64	46.25	38.95
SentiStrength	62.07	64.15	91.89	75.56	40.00	9.52	15.38	52.08	50.71	51.38
Senti-Median (SentiWordNet)	**81.03**	90.91	78.95	84.51	68.00	85.00	75.56	79.45	81.97	**80.03**
Senti-Median (MPQA)	77.59	90.00	72.97	80.60	64.29	85.71	73.47	77.14	79.34	77.03
Senti-Median (Thelwall-Lexicon)	79.31	84.85	80.00	82.35	72.00	78.26	75.00	78.42	79.13	78.68

Polarity Classification (Positive vs Negative)

Methods	Accuracy	Positive			Negative			Average		
		P	R	F1	P	R	F1	P	R	F1
MPQA-Method	72.5	80	92.31	85.71	71.43	45.45	55.56	75.71	68.88	70.63
SentiWordNet-Method	77.50	88.00	88.00	88.00	75.00	75.00	75.00	81.50	81.50	81.50
SentiStrength	85.00	95.65	81.48	88.00	70.59	92.31	80.00	83.12	86.89	84.00
Senti-Median (SentiWordNet)	**87.50**	89.29	92.59	90.91	83.33	76.92	80.00	86.31	84.76	**85.45**
Senti-Median (MPQA)	85.00	86.21	92.59	89.29	81.82	69.23	75.00	84.01	80.91	82.14
Senti-Median (Thelwall-Lexicon)	82.50	85.71	88.89	87.27	75.00	69.23	72.00	80.36	79.06	79.64

Table 6: Entity-level sentiment analysis results.

It can be observed from the upper panel of Table 6 that for subjectivity identification, our proposed Senti-Median method outperforms all the baselines by a large margin. In particular, merely using MPQA or SentiWordNet for sentiment labelling fails to detect any objective (neutral) entities. SentiStrength only achieves an F-measure of 15% for objective entity detection. On the contrary, our proposed Senti-Median method gives relatively balanced results on both subjective and objective identification. Senti-Median with the word prior sentiment obtained from SentiWordNet attains the best overall result with 81% in accuracy and 80% in F-measure, which outperforms the baselines by nearly 20% in accuracy and 30-40% in F-measure.

The lower panel of Table 6 shows the results of binary polarity identification (positive vs. negative) at the entity-level. SentiStrength performs better than MPQA-Method and SentiWordNet-Method, with 85% in accuracy and 84% in F-measure. Although our Senti-Median method with word prior sentiment obtained from either MPQA or Thelwall-Lexicon does not seem to bring any improvement over SentiStrength, Senti-Median based on SentiWordNet outperforms SentiStrength by 2.5% in accuracy and 1.5% in F-measure.

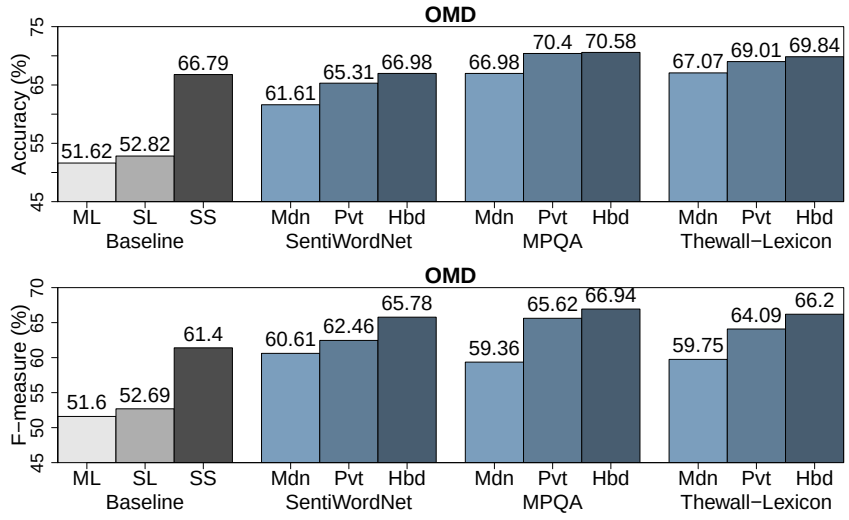
3.3.4.2 *Tweet-Level Sentiment Detection*

For tweet-level sentiment detection, we report the evaluation results using the Median method, the Pivot method and the Pivot-Hybrid method with SentiWordNet, MPQA and Thelwall lexicons on OMD, HCR and STS-Gold datasets. We also compare these results with those obtained from the baselines described in Section 4.2.3.3.

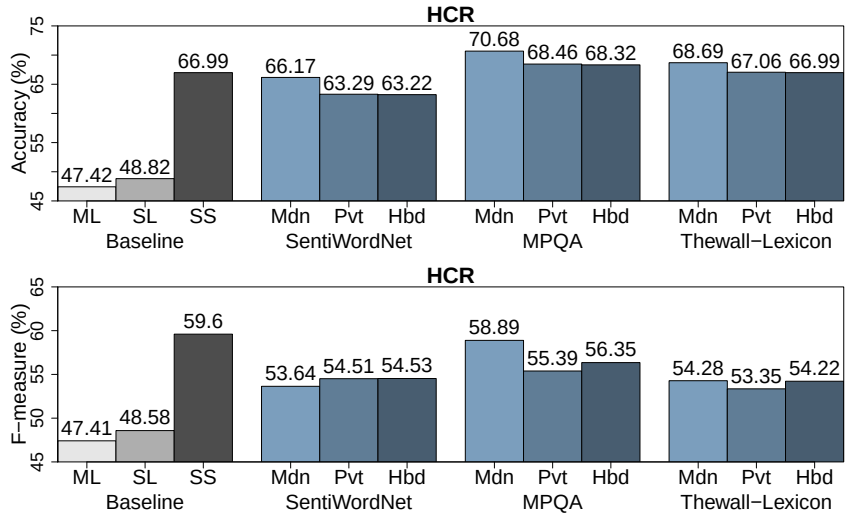
Figure 15 shows the results in accuracy (left column) and average F-measure (right column) of all the methods and across all the datasets. It can be observed that all our three methods outperform the MPQA-Method and SentiWordNet-Method baseline in both accuracy and average F-measure on all the datasets. On the OMD dataset, we observe a trend that Median $<$ Pivot $<$ Pivot-Hybrid. Both of our Pivot and Pivot-Hybrid methods give an average performance gain of 3.17% in accuracy and 4.3% in F-measure over SentiStrength with word prior sentiments obtained from either MPQA or Thelwall-Lexicon. The median method does not bring any performance gains over SentiStrength. Overall, the best result is achieved by our Hybrid-Pivot method with word prior sentiments obtained from MPQA. It outperforms SentiStrength by nearly 5% in accuracy and 6% in F-measure.

On the HCR dataset, all our three methods gave higher accuracy than SentiStrength with word prior sentiments obtained from either MPQA or Thelwall-Lexicon. In particular, the Median method coupled with MPQA outperforms SentiStrength by nearly 4% in accuracy. In terms of F-measure, The median based on MPQA gives a similar result as SentiStrength.

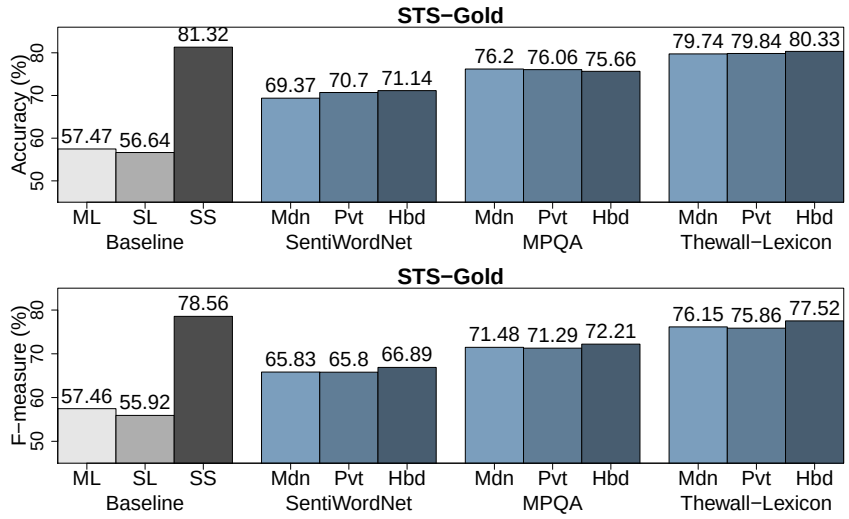
While the best sentiment classification accuracy on the OMD or HCR datasets is only about 70%, we managed to achieve 80% on the STS-Gold dataset in sentiment classification accuracy. We observe that the Pivot-Hybrid method outperforms both the Median and the Pivot methods regardless of where the word prior sentiments are obtained from for the STS-Gold dataset. Nevertheless, the Pivot-Hybrid method using Thelwall-Lexicon gives 1% lower accuracy and F-measure than SentiStrength which uses the same lexicon.



(a) Performance on OMD.



(b) Performance on HCR.



(c) Performance on STS-Gold.

Figure 15: Tweet-level sentiment detection results (Accuracy and F-measure), where ML: MPQA-Method, SL: SentiWordNet-Method, SS: SentiStrength, Mdn: SentiCircle with Median method, Pvt: SentiCircle with Pivot method, Hbd: SentiCircle with Pivot-Hybrid.

The above results show a close competition between our three SentiCircle methods and the SentiStrength method. The average accuracy of SentiCircle and SentiStrength across all three datasets is 72.39% and 71.7% respectively, and for F-measure it is 65.98% and 66.52%. Also, the average precision and recall for SentiCircle are 66.82% and 66.12%, and for SentiStrength are 67.07% 66.56% respectively.

Although the potential is evident, clearly there is a need for more research to determine the specific conditions under which SentiCircle performs better or worse. One likely factor that influences the performance of SentiCircle is the balance of positive to negative tweets in the dataset. For example, we notice that SentiCircle produces, on average, 2.5% lower recall than SentiStrength on positive tweet detection. This is perhaps not surprising since our tweet data contain more negative tweets than positive ones with the number of the former more than double the number of the latter (see Table 2).

3.3.4.3 *Impact on Words' Sentiment*

The motivation behind SentiCircle is that sentiment of words may vary with context. By capturing the contextual semantics of these words, using the SentiCircle representation, we aim to adapt the strength and polarity of words. We show here the average percentage of words in our three datasets for which SentiCircle changed their prior sentiment orientation or strength.

	SentiWordNet	MPQA	Thelwall-Lexicon	Average
Words found in the lexicon	54.86	16.81	9.61	27.10
Hidden words	45.14	83.19	90.39	72.90
Words flipped their sentiment orientation	65.35	61.29	53.05	59.90
Words changed their sentiment strength	29.30	36.03	46.95	37.43
New opinionated words	49.03	32.89	34.88	38.93

Table 7: Average percentage of words in three datasets, which their sentiment orientation or strength were updated by their SentiCircles.

Table 7 shows that on average 27.1% of the unique words in our datasets were covered by the sentiment lexicons and were assigned prior sentiments accordingly. Using the SentiCircle representation, however, resulted in 59.9% of these words flipping their sentiment orientations (e.g., from positive to negative, or to neutral) and 37.43% changing their sentiment strength while keeping their prior sentiment orientation. Hence only 2.67% of the words were left with their prior sentiment orientation and strength unchanged. It is also

worth noting that our model was able to assign sentiment to 38.93% of the *hidden* words that were not covered by the sentiment lexicons.

Our evaluation results showed that our SentiCircle representation coupled with the MPQA or Thelwall lexicons gives the highest performance amongst the other three lexicons. However, Table 7 shows that only 9.61% of the words in the three datasets were covered by the Thelwall-Lexicon, and 16.81% by MPQA. Nevertheless, SentiCircle performed best with these two lexicons, which suggests that it was able to cope with this low coverage by assigning sentiment to a large proportion of the *hidden* words.

3.4 SENTICIRCLES FOR ADAPTING SENTIMENT LEXICONS

In the previous section, we demonstrated the effectiveness of SentiCircles in capturing the words' contextual sentiment. Unlike SentiStrength, which uses a fixed sentiment score of words obtained from Thelwall-Lexicon, SentiCircles updates the prior sentiment (orientation and/or strength) of almost 50% words covered by Thelwall-Lexicon along the the three datasets. It also assigns contextual sentiment to 21.37% of words that are not covered by Thelwall-Lexicon. Therefore, our SentiCircle approach can be thought of as a lexicon-adaptation approach that takes as input a Twitter dataset and a sentiment lexicon (e.g., Thelwall-Lexicon), capture the words' context in tweets, and updates their sentiment orientation and strength in the lexicon accordingly.

In this section we continue with evaluating our SentiCircle approach against SentiStrength. However, instead of applying SentiCircles to directly detect the sentiment of entities or tweets, we propose using SentiCircles to amend the prior sentiment of words in Thelwall-Lexicon and then study the impact of doing so on the sentiment detection performance of SentiStrength. Remember that SentiStrength is a rule-based method that is specifically built to work with Thelwall-Lexicon. Therefore, measuring the performance of SentiStrength when using the adapted Thelwall-Lexicon allows us to further investigate and understand the role of contextual sentiment of individual words on the overall sentiment detection performance.

Figure 16 depicts the general two-step workflow for adapting sentiment lexicons with SentiCircles. First, we capture the contextual semantics and sentiment of the words in the given sentiment lexicon using SentiCircles as previously explained in Section 3.2.2.3. Secondly, a rule-based algorithm is applied to amend the prior sentiment of terms in the

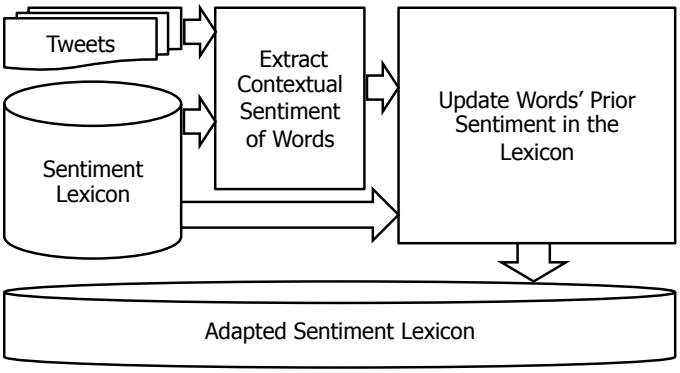


Figure 16: The systematic workflow of adapting sentiment lexicons with SentiCircles.

lexicon based on their corresponding contextual sentiment. We designed our rules with respect to the characteristics of Thelwall-Lexicon as will be subsequently explained.

Thelwall-Lexicon consists of 2546 terms coupled with integer values between -5 (very negative) and +5 (very positive). Based on the terms' prior sentiment orientations and strengths (SOS), we group them into three subsets of 1919 negative terms (SOS$\in$[-2,-5]), 398 positive terms (SOS$\in$[2,5]) and 229 neutral terms (SOS$\in${-1,1}).

Our adaptation method uses a set of antecedent-consequent rules that decides how the prior sentiment of the terms in Thelwall-Lexicon should be updated according to the positions of their SentiMedians (i.e., their contextual sentiment). In particular, for a term m, the method checks (i) its prior SOS value in Thelwall-Lexicon and (ii) the SentiCircle quadrant in which the SentiMedian of m resides. The method subsequently chooses the best-matching rule to update the term's prior sentiment and/or strength.

Table 8 shows the complete list of adaptation rules we use. As noted, these rules are divided into *updating rules*, i.e., rules for updating the existing terms in Thelwall-Lexicon, and *expanding rules*, i.e., rules for expanding the lexicon with new terms. The *updating rules* are further divided into rules that deal with terms that have similar prior and contextual sentiment orientations (i.e., both positive or negative), and rules that deal with terms that have different prior and contextual sentiment orientations (i.e., negative prior, positive contextual sentiment and vice versa).

Although they look complicated, the notion behind the proposed rules is rather simple: *Check how strong the contextual sentiment is and how weak the prior sentiment is* $\rightarrow$ *update the sentiment orientation and strength accordingly*. The strength of the contextual sentiment can be determined based on the sentiment quadrant of the SentiMedian of m, i.e., the contextual sentiment is strong if the SentiMedian resides in the "*Very Positive*" or "*Very*

Negative" quadrants (see Figure 11). On the other hand, the prior sentiment of m (i.e., $prior_m$) in Thelwall-Lexicon is weak if $|prior_m| \leqslant 3$ and strong otherwise.

Id	Antecedents	Consequent
Updating Rules (Similar Sentiment Orientations)		
1	$(\|prior\| \leqslant 3) \wedge (SentiMedian \notin StrongQuadrant)$	$\|prior\| = \|prior\| + 1$
2	$(\|prior\| \leqslant 3) \wedge (SentiMedian \in StrongQuadrant)$	$\|prior\| = \|prior\| + 2$
3	$(\|prior\| > 3) \wedge (SentiMedian \notin StrongQuadrant)$	$\|prior\| = \|prior\| + 1$
4	$(\|prior\| > 3) \wedge (SentiMedian \in StrongQuadrant)$	$\|prior\| = \|prior\| + 1$
Updating Rules (Different Sentiment Orientations)		
5	$(\|prior\| \leqslant 3) \wedge (SentiMedian \notin StrongQuadrant)$	$\|prior\| = 1$
6	$(\|prior\| \leqslant 3) \wedge (SentiMedian \in StrongQuadrant)$	$prior = -prior$
7	$(\|prior\| > 3) \wedge (SentiMedian \notin StrongQuadrant)$	$\|prior\| = \|prior\| - 1$
8	$(\|prior\| > 3) \wedge (SentiMedian \in StrongQuadrant)$	$prior = -prior$
9	$(\|prior\| > 3) \wedge (SentiMedian \in NeutralRegion)$	$\|prior\| = \|prior\| - 1$
10	$(\|prior\| \leqslant 3) \wedge (SentiMedian \in NeutralRegion)$	$\|prior\| = 1$
Expanding Rules		
11	$SentiMedian \in NeutralRegion$	$(\|contextual\| = 1) \wedge AddTerm$
12	$SentiMedian \notin StrongQuadrant$	$(\|contextual\| = 3) \wedge AddTerm$
13	$SentiMedian \in StrongQuadrant$	$(\|contextual\| = 5) \wedge AddTerm$

Table 8: Adaptation rules for Thelwall-Lexicon, where `prior`: prior sentiment value, `StrongQuadrant`: very negative/positive quadrant in the SentiCircle, `Add`: add the term to Thelwall-Lexicon.

For example, the word "`revolution`" in Thelwall-Lexicon has a weak negative sentiment (*prior*=-2) while it has a neutral contextual sentiment since its SentiMedian resides in the neutral region ($SentiMedian \in NeutralRegion$). Therefore, rule number 10 is applied and the term's prior sentiment in Thelwall lexicon will be updated to neutral ($|prior| = 1$). In another example, the words "`Obama`" and "`Independence`" are not covered by Thelwall-Lexicon, and therefore, they have no prior sentiment. However, their SentiMedians reside in the "*Positive*" quadrant in their SentiCircles, and therefore rule number 12 is applied and both terms will be assigned with a positive sentiment strength of 3 and added to the lexicon consequently.

3.4.1 Evaluating SentiStrength on the Adapted Thelwall-Lexicon

As mentioned earlier, our goal behind adapting Thelwall-Lexicon is to demonstrate the importance of context captured by our SentiCircles model on the sentiment orientation and strength of words, and consequently on the overall sentiment detection performance of SentiStrength when uses the adapted lexicon.

Our evaluation starts with adapting Thelwall-Lexicon under three different settings: (i) the *update* setting where we update the prior sentiment of *existing terms* in the lexicon, (ii) the *expand* setting where we expand Thelwall-Lexicon with *new opinionated terms*, and (iii) the *update+expand* setting where we try both settings together. After that, we use SentiStrength with each of the adapted lexicons to perform tweet-level binary polarity classification (positive vs. negative) on OMD, HCR and STS-Gold datasets (see Table 2).

Applying our adaptation approach to Thelwall-Lexicon results in substantial changes in it. Table 9 shows the percentage of words in the three datasets that were found in Thelwall-Lexicon with their sentiment changed after adaptation. One can notice that on average 9.61% of the words in our datasets were found in the lexicon. However, updating the lexicon with the contextual sentiment of words resulted in 33.82% of these words flipping their sentiment orientation and 62.94% changing their sentiment strength while keeping their prior sentiment orientation. Only 3.24% of the words in Thelwall-Lexicon remained untouched. Moreover, 21.37% of words previously unseen in the lexicon were assigned with contextual sentiment by our approach and added to Thelwall-Lexicon subsequently.

Note that the numbers in Table 7 that reflect the impact on Thelwall-Lexicon are sightly different to the average impact's numbers we report in Table 9. This is because our adaptation rules in this study consider both, the prior and contextual sentiment of words when adapting their sentiment in the lexicon as explained earlier. According to our rules, words' prior sentiment do not need to be always overridden with their contextual sentiment. On the other hand, in our study in Section 3.3.1 we only consider the contextual sentiment of words, i.e., we always override the prior sentiment of words in the lexicon with their contextual ones.

Table 10 shows the average results of binary sentiment classification performed by SentiStrength on our datasets using (i) the original Thelwall-Lexicon (*Original*), (ii) Thelwall-Lexicon adapted under the *update* setting (*Updated*), and (iii) Thelwall-Lexicon adapted

	OMD	HCR	STS-Gold	Average
Words found in the lexicon	12.43	8.33	8.09	9.61
Hidden words	87.57	91.67	91.91	90.39
Words flipped their sentiment orientation	35.02	35.61	30.83	33.82
Words changed their sentiment strength	61.83	61.95	65.05	62.94
Words remained unchanged	3.15	2.44	4.13	3.24
New opinionated words	23.94	14.30	25.87	21.37

Table 9: Average percentage of words in the three datasets that had their sentiment orientation or strength updated by our adaptation approach.

under the *update+expand* setting.[11] The table reports the results in accuracy and three sets of precision (P), recall (R), and F-measure (F1), one for positive sentiment detection, one for negative, and one for the average of the two.

From these results in Table 10, we notice that the best classification performance in accuracy and F1 is obtained on the STS-Gold dataset regardless the lexicon being used. We also observe that the negative sentiment detection performance is always higher than the positive detection performance for all datasets and lexicons.

Datasets	**Lexicons**	**Accuracy**	**Positive Sentiment**			**Negative Sentiment**			**Average**		
			P	R	F1	P	R	F1	P	R	F1
OMD	Original	66.79	55.99	40.46	46.97	70.64	81.83	75.82	63.31	61.14	61.4
	Updated	**69.29**	58.89	51.4	54.89	74.12	79.51	76.72	66.51	65.45	65.8
	Updated+Expanded	69.2	58.38	53.18	55.66	74.55	78.34	76.4	66.47	65.76	**66.03**
HCR	Original	66.99	43.39	41.31	42.32	76.13	77.64	76.88	59.76	59.47	**59.6**
	Updated	**67.21**	42.9	35.77	39.01	75.07	80.25	77.58	58.99	58.01	58.29
	Updated+Expanded	66.99	42.56	36.02	39.02	75.05	79.83	77.37	58.8	57.93	58.19
STS-Gold	Original	81.32	68.75	73.1	70.86	87.52	85.02	86.25	78.13	79.06	78.56
	Updated	81.71	69.46	73.42	71.38	87.7	85.45	86.56	78.58	79.43	78.97
	Updated+Expanded	**82.3**	70.48	74.05	72.22	88.03	86.02	87.01	79.26	80.04	**79.62**

Table 10: Cross comparison results of original and the adapted lexicons.

As for different lexicons, we notice that on OMD and STS-Gold the adapted lexicons outperform the original lexicon in both accuracy and F-measure. For example, on OMD

11 Note that in this work we do not report the results obtained under the expand setting since no improvement was observed comparing to the other two settings.

the adapted lexicon shows an average improvement of 2.46% and 4.51% in accuracy and F1 respectively over the original lexicon. On STS-Gold the performance improvement is less significant than that on OMD, but we still observe 1% improvement in accuracy and F1 compared to using the original lexicon. As for the HCR dataset, the adapted lexicon gives on average similar accuracy, but 1.36% lower F-measure. This performance drop can be attributable to the poor detection performance of positive tweets. Specifically, we notice from Table 10 a major loss in the recall on positive tweet detection using both adapted lexicons. One possible reason is the sentiment class distribution in our datasets. In particular, one may notice that HCR is the most imbalanced amongst the three datasets. Moreover, by examining the numbers in Table 7, we can see that HCR presents the lowest number of new opinionated words among the three datasets (i.e., 10.61% lower than the average) which could be another potential reason for not observing any performance improvement.

3.5 RUNTIME ANALYSIS

In this section, we perform runtime analysis of the various methods and algorithms that constitute our SentiCircle model. To this end, we apply our model to the STS-Gold dataset using a computer with a i7 core CPU 2.3GHz and 8G memory. According to the results reported in Table 11, building a term-index out of 2034 tweets (5035 unique terms) takes 13.8 seconds. This is not surprising given that this task involves several subtasks (tokenization, part-of-speech tagging, negation, Non-English text filtering). We also construct context-term vectors and extract the terms' contextual features during this task as they all happen together. Although 13.8 seconds sounds relatively slow, building a term-index for a tweet collection is a one-time task, i.e., once it is built, the term-index can be used to perform various tasks in a relatively short time. For example, composing a SentiCircle from the term-index takes only 47.18 milliseconds. Moreover, the term-index can be updated with new tweets on the fly, where it takes 6.6 milliseconds to add a new tweet to the index.

We also estimate the runtime of our sentiment detection methods at entity and tweet levels. For entity-level, calculating Senti-Median takes 10 milliseconds per entity. For tweet-level, the Median method takes 16.9 milliseconds per tweet while the Pivot and Pivot-Hybrid methods take 0.01 and 4.53 milliseconds respectively.

Task	Run Time
Term-Index(Generate)	13.8 second
Term-Index(Update)	6.6 millisecond
SentiCircle	47.18 millisecond
Senti-Median Method	10 millisecond
Median Method	16.9 millisecond
Pivot Method	0.01 millisecond
Pivot-Hybrid Method	4.53 millisecond

Table 11: Runtime analysis of the SentiCircle model on the STS-Gold dataset.

3.6 DISCUSSION

We demonstrated the value of using contextual semantics in sentiment analysis on Twitter. In particular, we proposed a semantic representation model called SentiCircle for capturing words' contextual semantics in tweets data. We studied the effectiveness of SentiCircle in (i) lexicon-based sentiment analysis at both, entity- and tweet-level and (ii) sentiment lexicon adaptation.

SentiCircle for Lexicon-based Sentiment Analysis: We proposed using SentiCircle to perform lexicon-based sentiment detection at the tweet and entity levels.

At the entity level, the evaluation was done on a single dataset (STS-Gold) that we constructed ourselves due to the sheer lack of gold-standard datasets for evaluating entity-level sentiment.

We have seen that merely using MPQA or SentiWordNet for sentiment labelling fails to detect any neutral entities. This is expected since words in both lexicons are oriented with positive and negative scores, but not with neutral ones. SentiCircle, on the other hand, was able to amend sentiment scores of words in both lexicons based on contexts - hence achieving a much higher performance in detecting neutral entities than all the baselines. The next future step in this direction would be to compare our approach against more sophisticated methods including machine learning ones, such as those based on the probabilistic distribution of words' sentiment in a given Twitter corpus [12], or SVM classifiers [77]. Another future direction is to experiment with our approach using new

datasets of entities of different topic foci, which allows for better understanding of the settings/conditions under which our approach produces better or worse performances.

At the tweet level, the evaluation was performed on three Twitter datasets and using three different sentiment lexicons. The results showed that our SentiCircle approach outperforms significantly MPQA-Method and SentiWordNet-Method. Compared to SentiStrength, the results were not as conclusive, since SentiStrength slightly outperformed SentiCircles on the STS-Gold dataset, and also yielded marginally better F-measure for the HCR dataset. This might be due to the different topic distribution in the datasets. The STS-Gold dataset contains random tweets, with no particular topic focus, whereas OMD and HCR consist of tweets that discuss specific topics, and thus the contextual semantics extracted by SentiCircle are presumably more representative in these datasets than in STS-Gold. Other important characteristics could be the sparseness degree of data and the positive and negative distribution of tweets. Investigating these issues and their influence on the performance of our approach creates room for more future work.

We extracted opinion targets (Pivot terms) in the Pivot-Method by looking at their POS-tags assuming that all pivot terms in a given tweet receive similar sentiment. Another direction for future work is to refine the pivot extraction method to consider cases, where the tweet contains several pivot terms of different sentiment orientations.

We proposed and tested methods that assign positive, negative or neutral sentiment to terms and tweets based on their corresponding SentiCircle representations. However, there could be a need for considering cases where terms with "`Mixed`" sentiment emerge, when their SentiCircle representations consist of positive and negative terms only.

SentiCircles for Adapting Sentiment Lexicons: In the second part of this chapter, we showed the potential of using contextual semantics of words for adapting general-purpose sentiment lexicons. As a case study, we used SentiCircles to adapt the prior sentiment of words in Thelwall-Lexicon with respect to their contextual sentiment.

The adapted Thelwall-Lexicon outperformed the original lexicon in two out of three datasets. However, a performance drop was observed in the HCR dataset using our adapted lexicons. Our initial observations suggest that the quality of our approach might be dependent on the sentiment class distribution in the dataset. Therefore, a deeper investigation in this direction is required.

Our adaptation rules are specific to Thelwall-Lexicon. These rules, however, can be generalized to other lexicons, which constitutes another future direction of this work.

3.7 SUMMARY

Contextual semantics refer to the type of semantics that is extracted from the co-occurrences of words in texts. In this chapter, we presented our work on using the contextual semantics of words in sentiment analysis on Twitter, aiming at addressing the first research question of this thesis: *Could the contextual semantics of words enhance lexicon-based sentiment analysis performance?*.

To this end, we first proposed a novel contextual semantic sentiment representation of words, called SentiCircle, which is able to assign sentiment to words with respect to their contextual semantics in tweets. After that, we tested the effectiveness of our proposed representation in three sentiment analysis tasks on Twitter; entity-level sentiment analysis, tweet-level sentiment analysis, and sentiment lexicon adaptation.

For entity- and tweet-level sentiment analysis tasks, we proposed several lexicon-based methods, based on SentiCircles, to detect the sentiment of individual entities as well the overall sentiment of the tweets these entities occur within. We evaluated and tested our methods under different settings (three different sentiment lexicons and three different datasets) and compared their performance against various lexicon baseline methods. For entity-level sentiment detection, our results showed that our proposed method based on SentiCircles outperforms all the other methods by a large margin in both, accuracy and in F-measure. For tweet-level sentiment detection, our methods overtake the state-of-the-art SentiStrength method in accuracy, but fall marginally behind in F-measure.

As for sentiment lexicon adaptation task, we proposed a rule-based approach that employs SentiCircles to amend the prior sentiment of words in a given general-purpose sentiment lexicon with respect to their contextual one. The evaluation was done on Thelwall-Lexicon using three Twitter datasets. Results showed that lexicons adapted by our approach improved the sentiment classification performance over the the original Thelwall-Lexicon, in both accuracy and F1 on two out of three Twitter datasets.

4

CONCEPTUAL SEMANTICS FOR SENTIMENT ANALYSIS OF TWITTER

After demonstrating in the previous chapter the effectiveness of contextual semantics for computing words' and tweets' sentiment, in this chapter we introduce our work on conceptual semantics and their use in sentiment analysis on Twitter. In essence, we propose several methods for incorporating the semantic concepts of named entities, extracted from tweets into both, supervised and lexicon-based sentiment analysis approaches. Our findings suggest that conceptual semantics, when used as features, increase the performance of tweet-level sentiment analysis in comparison with features extracted from the syntactic, linguistic, or statistic representation of words.

4.1 INTRODUCTION

As discussed earlier in this thesis, traditional sentiment analysis approaches on Twitter are semantically weak since they do not account for the latent semantics of words when calculating their sentiment in the tweets [33, 139].

In chapter 3 we showed that semantics extracted based on the context of words (contextual semantics) can be effectively used to capture words' sentiment and consequently improve the performance of lexicon-based sentiment analysis approaches. In this chapter, we experiment with the use of conceptual semantics for sentiment analysis. Conceptual semantics is often extracted from external knowledge sources, such as ontologies and semantic networks [33].

The research question we aim to address in this chapter is:

RQ2 ***Could the conceptual semantics of words enhance sentiment analysis performance?***

To address the above question, we propose extracting and incorporating the conceptual semantics of words into both supervised and lexicon-based approaches to enhance their performances in tweet-level sentiment analysis. For the supervised machine approach, we investigate three methods of incorporating conceptual semantics as features to train

supervised sentiment classifiers. For the lexicon-based approach, we consider augmenting conceptual semantics into our previously proposed model, SentiCircle (see Chapter 3) and study the impact of such incorporation on the overall sentiment detection performance.

Our evaluation is conducted by analysing the performance of various sentiment analysis methods in tweet-level sentiment analysis, after incorporating conceptual semantics. To this end, we use three different Twitter datasets and compare against several state-of-the-art baselines.

Results show that, when incorporating conceptual semantics in machine learning approaches, the performance improves up to 4% in F1-measure on average over models trained from syntactic features solely.

For lexicon-based sentiment analysis, enriching SentiCircle with conceptual semantics improves the performance our lexicon-based approach, by 0.39% in average F1-measure, but gives 0.19% lower accuracy, in comparison with using SentiCircle with no semantic enrichment.

Lastly, our results show that the advantage of using conceptual semantics over other techniques is mostly restricted to negative sentiment identification, in large topically-diverse datasets.

The rest of this chapter is organised as follows. Section 4.2 introduces our approach of extracting and using conceptual semantics with supervised sentiment analysis approaches. Section 4.3 demonstrates how to enrich SentiCircles with conceptual semantics to perform lexicon-based sentiment sentiment analysis. Discussion is delivered in Section 4.4. Finally, we summarise our work and findings in this chapter in Section 4.5.

4.2 CONCEPTUAL SEMANTICS FOR SUPERVISED SENTIMENT ANALYSIS

In this section, we describe our approach of extracting and exploiting conceptual semantics as features for sentiment classifier training in order to improve the overall sentiment classification performance.

We have seen earlier in Chapter 2 that the sparsity of Twitter data affects, to a large extent, the performance of supervised sentiment classifiers [137]. Due to sparsity, many terms in the test set of a classifier do not occur in its training set, preventing therefore, the classifier from detecting their sentiment. To address this problem, the semantic concepts of entities extracted from tweets can be used to measure the overall correlation of a group of entities (e.g. all Apple products) with a given sentiment polarity. Hence adding semantic concepts as features to the analysis could help identify the sentiment of tweets that contain any of the entities that such concepts represent, even if those entities never appeared in the training set (e.g. a new gadget from Apple).[1]

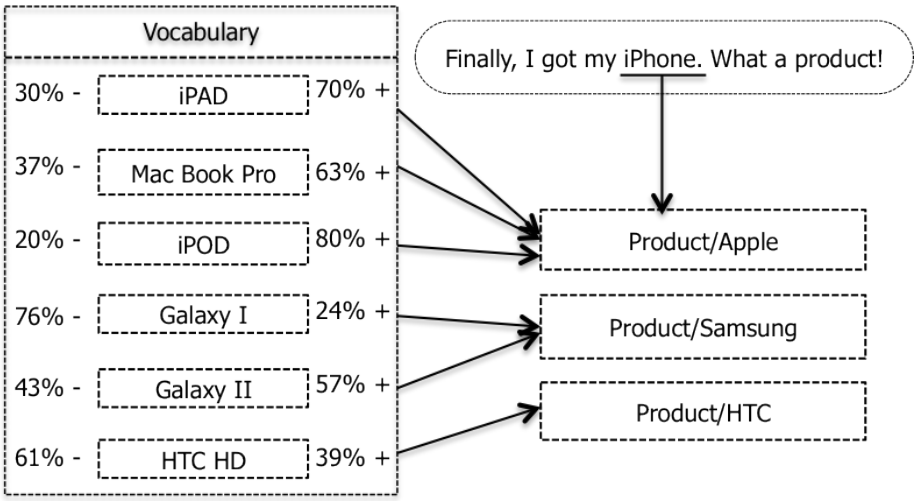


Figure 17: Measuring correlation of semantic concepts with negative/positive sentiment. These semantic concepts are then incorporated in sentiment classification.

An example showing the usefulness of using semantic concepts as features for sentiment classifier training is depicted in Figure 17, where the left box lists entities appearing in the training set, together with their occurrence probabilities in positive and negative tweets. For example, the entities "*iPad*", "*iPod*" and "*Mac Book Pro*" appear more often

1 Assuming of course that the entity extractor successfully identify the new entities as sub-types of concepts already correlated with negative or positive sentiment.

in tweets of positive polarity and they are all mapped to the semantic concept PRODUCT/APPLE. As a result, the tweet from the test set *"Finally, I got my iPhone. What a product!"* is more likely to have a positive polarity because it contains the entity *"iPhone"* which does not occur in the training set, but is mapped to the concept PRODUCT/APPLE.

Based on the above description, our proposed pipeline of adding semantic concepts into supervised sentiment classification methods breaks down into two main steps, as depicted in Figure 18:

- Extraction: this step aims to detect named-entities in a given tweet corpora and to extract their associated semantic concepts using third-party entity extraction tools (Section 4.2.1).
- Incorporation: The extracted semantic concepts are incorporated as features into the feature space of a given supervised machine learning classifier. To this end, we propose three different incorporation methods; by replacement, by augmentation, and by interpolation (Section 4.2.2). The output of this step is a semantic sentiment classifier which can be used to infer the sentiment of a given collection of unlabelled tweets (Section 4.2.4).

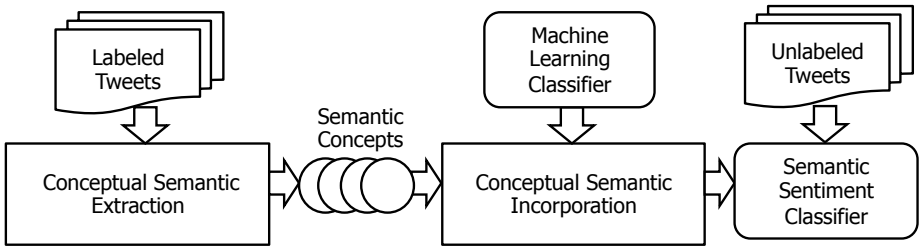


Figure 18: Systematic workflow for exploiting conceptual semantics in supervised sentiment classification on Twitter.

4.2.1 Extracting Conceptual Semantics

There are several open APIs that provide entity extraction services for online textual data. Rizzo and Troncy [132] evaluated the use of five popular entity extraction tools on a dataset of news articles, including AlchemyAPI,[2] DBPedia Spotlight,[3] Extractiv,[4] Ze-

2 www.alchemyapi.com
3 http://dbpedia.org/spotlight/
4 http://wiki.extractiv.com/w/page/29179775/Entity-Extraction

manta,[5] and OpenCalais.[6] Their experimental results showed that AlchemyAPI performs best for entity extraction and semantic concept mapping. Our datasets consist of informal tweets, and hence are intrinsically different from those used in [132]. Therefore we conducted our own evaluation, and randomly selected 500 tweets from the STS corpus [60] and asked 3 evaluators to assess the semantic concept extraction outputs generated from AlchemyAPI, OpenCalais and Zemanta.

	No. of Concepts	Entity-Concept Mapping Accuracy (%)		
Extraction Tool	Extracted	Evaluator 1	Evaluator 2	Evaluator 3
AlchemyAPI	108	73.97	73.8	72.8
Zemanta	70	71	71.8	70.4
OpenCalais	65	68	69.1	68.7

Table 12: Evaluation results of AlchemyAPI, Zemanta and OpenCalais.

The assessment of the outputs was based on (1) the correctness of the extracted entities (i.e., is the retrieved entity is a real-world entity or not?); and (2) the correctness of the entity-concept mappings (i.e., is the extracted entity correctly mapped to its right concept or not?). The evaluation results presented in Table 12 show that AlchemyAPI extracted the most number of concepts and it also has the highest entity-concept mapping accuracy compared to OpenCalais and Zemanta. As such, we chose AlchemyAPI to extract the semantic concepts from the Twitter datasets we use in our experiments.

4.2.2 Conceptual Semantics Incorporation

In this section we describe how to use conceptual semantic features to train supervised machine learning classifiers. In particular, we propose three different methods to incorporate semantic features into Naive Bayes (NB) classifier training. We start by an overview of the NB followed by our proposed incorporation methods. Our choice of using Naive Bayes as a case study in this work is due to its common and effective use as a sentiment classifier in the literature on Twitter sentiment analysis (see Chapter 2). Moreover, Naive Bayes provides a simple, yet efficient language model that is easy to train, extend and implement [18].

5 www.zemanta.com
6 www.opencalais.com

NB is a probabilistic classifier, where the assignment of a sentiment class c (positive or negative) to a given tweet **w** can be computed as:

$$\begin{aligned}\hat{c} &= \arg\max_{c \in \mathcal{C}} P(c|\mathbf{w}) \\ &= \arg\max_{c \in \mathcal{C}} P(c) \prod_{1 \leqslant i \leqslant N_{\mathbf{w}}} P(w_i|c),\end{aligned} \tag{10}$$

where $N_{\mathbf{w}}$ is the total number of words in tweet **w**, $P(c)$ is the prior probability of a tweet appearing in class c, $P(w_i|c)$ is the conditional probability of word w_i occurring in a tweet of class c.

In multinomial NB, $P(c)$ can be estimated by $P(c) = N_c/N$ Where N_c is the number of tweets in class c and N is the total number of tweets. $P(w_i|c)$ can be estimated using maximum likelihood [142] with Laplace smoothing [95]:

$$P(w|c) = \frac{N(w,c)+1}{\sum_{w' \in V} N(w'|c) + |V|} \tag{11}$$

where $N(w,c)$ is the occurrence frequency of word w in all training tweets of class c and $|V|$ is the number of words in the vocabulary.

To incorporate semantic concepts into NB learning, we propose three different methods as described below.

Semantic Replacement: In this method, we replace all entities in tweets with their corresponding semantic concepts (e.g., "`Headache`" will be replaced in all tweets by its associated concept "`Health Condition`"). This leads to the reduction of the vocabulary size, where the new size is determined by:

$$|V'| = |V| - |W_{\text{entity}}| + |S| \tag{12}$$

where $|V'|$ is the new vocabulary size, $|V|$ is the original vocabulary size, $|W_{\text{entity}}|$ is the total number of unique entity words that have been replaced by the semantic concepts, and $|S|$ is the the total number of semantic concepts associated to the previously identified entities.

Semantic Augmentation: This method augments the original feature space with the semantic concepts as additional features for the classifier training (e.g., "`Headache`" and its concept "`Health Condition`" will appear together in the feature space). The size of the vocabulary in this case is enlarged by the semantic concepts introduced:

$$|V'| = |V| + |S| \tag{13}$$

Semantic Interpolation: In this method, we interpolate the unigram language model in NB with the generative model of words given semantic concepts. We propose a general interpolation method below which is able to interpolate an arbitrary type of features such as semantic concepts, POS sequences, sentiment-topics, etc.

Thus, the new language model with interpolation has the following formula:

$$P_f(W|C) = \alpha\, P_u(W|C) + \sum_i \beta_i P(W, F_i, C) \tag{14}$$

Where $P_f(W|C)$ is the new language model with interpolation, $P_u(W|C)$ is the original unigram class model and can be calculated using the maximum likelihood estimation, $P(W, F_i, C)$ is the interpolation component, and F_i is a feature vector of type i (e.g., semantic concepts). The coefficients α and β_i are used to control the influence of the interpolated features in the new language model where:

$$\alpha + \sum_i \beta_i = 1$$

By setting α to 1 the class model becomes a unigram language model without any feature interpolation. On the other hand, setting α to 0 reduces the class model to a feature mapping model. In this work, values of these coefficients have been set by conducting a sensitivity test on three Twitter corpora as will be discuss in Section 4.2.4.1.

The interpolation component in the equation 14 can be decomposed as follows:

$$P(W, F_i, C) = \sum_j P(W|f_{ij})P(f_{ij}|C) \tag{15}$$

Where f_{ij} is the j-th feature of type i, $P(f_{ij}|C)$ is the distribution of features f_{ij} in the training data given the class C and $P(W|f_{ij})$ is the distribution of words in the training data given the feature f_{ij}. Both distributions can be computed via the maximum likelihood estimation.

4.2.3 Evaluation Setup

Our proposed approach, as shown in the previous sections, extracts conceptual semantics from Twitter data and uses them as features to train NB classifiers using three different methods. We assess the performance our approach for tweet-level sentiment analysis using three different Twitter datasets and compare against models trained from other three types of features, as will be explained subsequently.

4.2.3.1 *Datasets*

We assess the use of conceptual semantics in supervised sentiment classification using three different Twitter datasets: STS-Expand, HCR and OMD. The statistics of the datasets are shown in Table 13. Note that the OMD and the HCR datasets contain tweets about specific topics: the US Health Care Reform bill in the HCR dataset, and the Obama McCain debate in the OMD dataset. On the other hand, the STS-Expand consists of general tweets with no topic focus. We refer the reader to the body of Appendix A for a full description of the construction and annotation of these datasets.

Dataset	Type	No. of Tweets	Positive	Negative
STS-Expand [137]	Train	60K	30K	30K
	Test	1,000	470	530
HCR [152]	Train	655	234	421
	Test	699	163	536
OMD [47]	n-fold cross validation	1,081	393	688

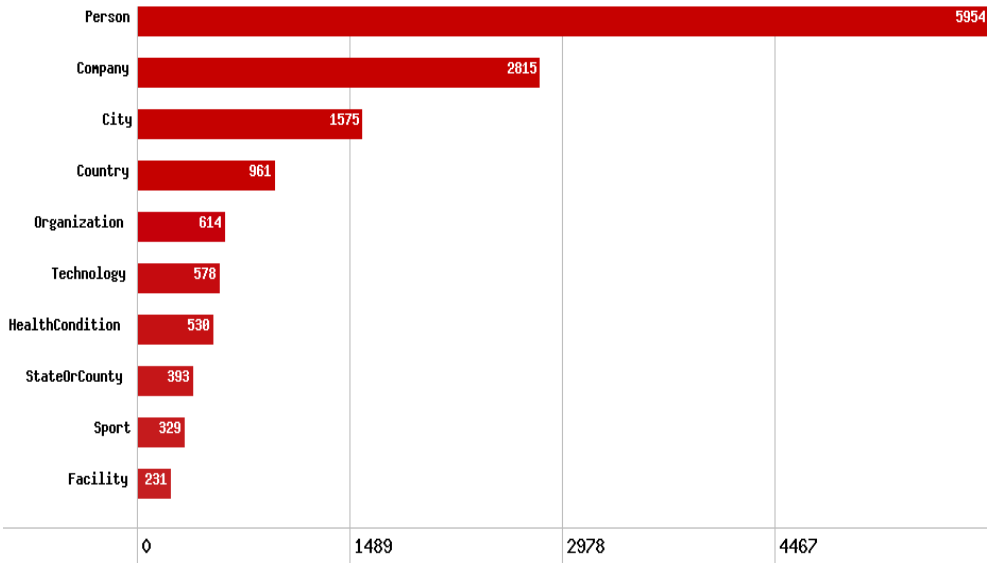
Table 13: Statistics of the three Twitter datasets used in this paper.

4.2.3.2 *Semantic Concepts Extraction*

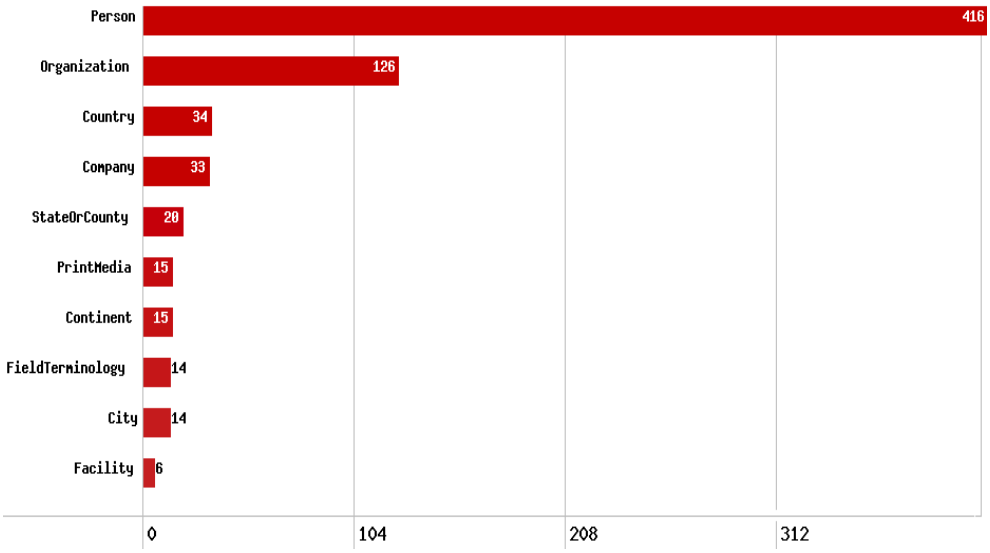
As mentioned in Section 4.2.1, we use AlchemyAPI to extract the semantic concepts from our three datasets. Figure 19 shows the top ten high-level extracted concepts from the three datasets with the number of entities associated with each of concept. It can be observed that the most frequent semantic concept is `Person` across all the three corpora. The next two most frequent concepts are `Company` and `City` for STS-Expand, `Organisation` and `Country` for HCR, and `Country` and `Company` for OMD. The level of specificity of these concepts (i.e., high-level vs. low-level concepts) is determined by AlchemyAPI. Table 14 lists the total number of entities extracted and the number of semantic concepts mapped against them for each dataset.

4.2.3.3 *Baselines*

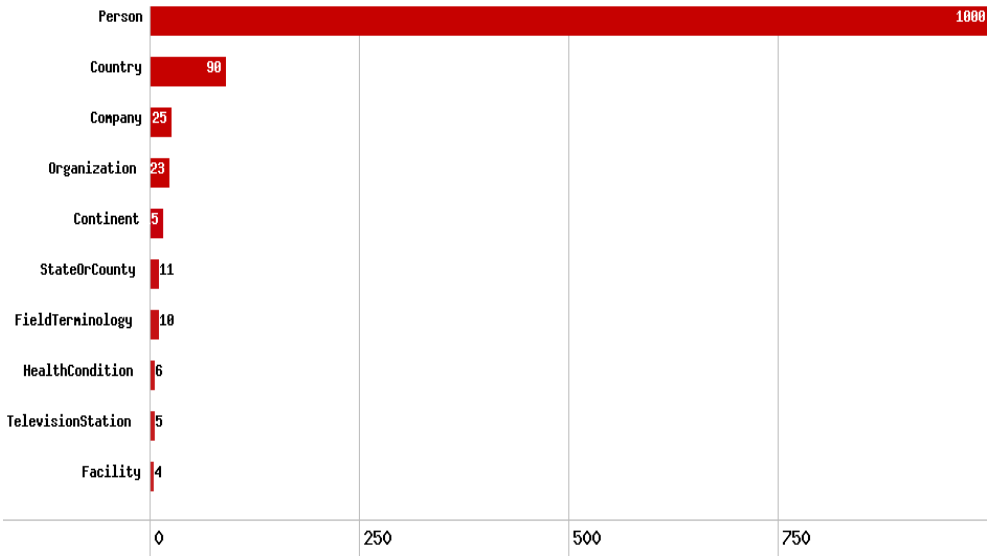
We compare the performance of our semantic sentiment analysis approach against the baselines described below.



(a) STS-Expand.



(b) HCR.



(c) OMD.

Figure 19: Top 10 frequent concepts extracted with the number of entities associated with them.

	STS-Expand	HCR	OMD
No. of Entities	15139	1392	1194
No. of Concepts	29	19	14

Table 14: Entity/concept extraction statistics of STS-Expand, OMD and HCR using AlchemyAPI.

Unigrams Features: Word unigrams are among the simplest types of features used for sentiment analysis of Twitter data. Models trained from word unigrams were shown to outperform random classifiers by a decent margin of 20% [1]. In this work we use NB classifiers trained from word unigrams as our first baseline model. Table 15 lists, for each dataset, the total number of the extracted unigram features that are used for the classification training.

Dataset	No. of Unigrams
STS-Expand	37054
HCR	2060
OMD	2364

Table 15: Total number of unigram features extracted from each dataset.

Part-of-Speech Features: POS features are common features that have been widely used in the literature for the task of Twitter sentiment analysis. In this work, we build various NB classifiers trained using a combination of word unigrams and POS features and use them as baseline models. We extract the POS features using the TweetNLP POS tagger [109],[7] which is trained specifically from tweets. This differs from the previous work [110, 1], which relies on POS taggers trained from treebanks in the newswire domain for POS tagging. It was shown that the TweetNLP tagger outperforms the Stanford tagger[8] with a relative error reduction of 25% when evaluated on 500 manually annotated tweets [59]. Moreover, the tagger offers additional recognition capabilities for abbreviated phrases, emoticons and interjections (e.g. “lol", “omg").

Sentiment-Topic Features: The sentiment-topic features (JST features) denote the latent topics (aka groups or patterns) and the topic-associated sentiment in texts. They are

7 http://www.ark.cs.cmu.edu/TweetNLP/

8 http://nlp.stanford.edu/software/tagger.shtml

extracted from tweets using the weakly-supervised joint sentiment-topic (JST) model [87]. JST is a four-layer generative model, which allows the detection of both, sentiment and topic simultaneously, from text based on the statistical similaries (contextual semantic similarities) between words (see Chapter 2).

To extract JST features, we trained the JST model on the training set with tweet sentiment labels discarded. The resulting model assigns each word in tweets with a sentiment label and a topic label. Hence JST essentially groups different words that share similar sentiment and topic.

We list some of the topic words extracted by this model from the STS-Expand and OMD corpora in Table 16. Words in each cell are grouped under one topic. The upper half of the table shows topic words bearing positive sentiment while the lower half shows topic words bearing negative sentiment. For example, Topic 2 under positive sentiment is about the movie "Twilight", while Topic 5 under negative sentiment is about a complaint of feeling sick, possibly due to cold and headache. The rational behind this model is that grouping words under the same topic and bearing similar sentiment could reduce data sparseness in Twitter sentiment classification and improve accuracy.

Algorithm 1 shows how to perform NB training with sentiment-topics extracted from JST. The training set consists of labeled tweets, $\mathcal{D}^{train} = \{(\mathbf{w}_n; c_n) \in \mathcal{W} \times \mathcal{C} : 1 \leqslant n \leqslant N^{train}\}$, where $\mathcal{W}$ is the input space and $\mathcal{C}$ is a finite set of class labels. The test set contains tweets without labels, $\mathcal{D}^{test} = \{\mathbf{w}_n^t \in \mathcal{W} : 1 \leqslant n \leqslant N^{test}\}$. A JST model is first learned from the training set and then a topic and a sentiment are inferred for each tweet in the test set. The original tweets are augmented with those sentiment-topics, as shown in Step 4 of Algorithm 1, where $l_i\_z_i$ denotes a combination of sentiment label l_i and topic z_i for word w_i in a tweet $\mathbf{w}_n$. Finally, an optional feature selection step can be performed according to Information Gain (IG) and a classifier is then trained from the training set with the new feature representation.

4.2.4 Evaluation Results

In this section we evaluate the use of the sentiment features discussed in 4.2 and present the sentiment identification results on the STS-Expand, HCR and OMD datasets. We then compare these results with those obtained from using the baseline features described in Section 4.2.3.3.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Positive	win	twilight	make	tomorrow	today
	final	movie	mccain	weekend	nice
	watch	award	debate	school	Sunday
	game	moon	good	start	enjoy
	luck	tonight	point	plan	weather
	today	watch	interest	fun	love
	week	mtv	right	yai	walk
	hope	excited	answer	wait	sunny
Negative	iphone	dog	obama	miss	feel
	internet	sad	question	far	sick
	download	death	understand	travel	bad
	apple	accident	doesn't	mum	hurt
	store	today	answer	away	pain
	slow	car	comment	dad	flu
	issue	awful	back	love	sore
	crash	cry	debate	country	horrible

Table 16: Extracted sentiment-topic words by the sentiment-topic model (JST) [87].

Algorithm 1 NB training with sentiment-topics extracted from JST.

Input: The training set $\mathcal{D}^{train}$ and test set $\mathcal{D}^{test}$

Output: NB sentiment classifier

1: Train a JST model on $\mathcal{D}^{train}$ with the tweet sentiment labels discarded
2: Infer sentiment-topic for $\mathcal{D}^{test}$
3: **for** each tweet $\mathbf{w}_n = (w_1, w_2, ..., w_m) \in \{\mathcal{D}^{train}, \mathcal{D}^{test}\}$ **do**
4: Augment tweet with sentiment-topics generated from JST,
$\mathbf{w}'_n = (w_1, w_2, ..., w_m, l_1\_z_1, l_2\_z_2, ..., l_m\_z_m)$
5: **end for**
6: Create a new training set $\mathcal{D}^{train'} = \{(\mathbf{w}'_n; c_n) : 1 \leqslant n \leqslant N^{train}\}$
7: Create a new test set $\mathcal{D}^{test'} = \{\mathbf{w}'_n : 1 \leqslant n \leqslant N^{test}\}$
8: Perform feature selection using IG on $\mathcal{D}^{train'}$
9: Return NB trained on $\mathcal{D}^{train'}$

We use NB trained from word unigrams as the starting-point baseline model. The features are incorporated into NB by either, the interpolation approach described in Section 4.2.2, or by simply augmenting the original bag-of-words feature space. For evaluation on STS-Expand and HCR, we use the training and testing sets shown in Table 13. For OMD, we perform 5-fold cross validation and report the results averaged over 10 runs. This is because unlike STS-Expand and HCR, the OMD dataset is published as one unpartitioned collection of tweets.

The raw tweets data can be very noisy, and hence some pre-processing was necessary, such as replacing all hyperlinks with "*URL*", converting some words with apostrophe, such as "*hate'n*", to their complete form "*hating*", removing repeated letters (e.g. "*loovee*" becomes "*love*"), and removing words that have less than 3 letters.

4.2.4.1 *Results on Incorporating Semantic Features*

Semantic features can be incorporated into NB training in three different ways, *replacement*, *augmentation*, and *interpolation* (Section 4.2.2). Table 17 shows the F measures produced when using each of these feature incorporation methods. With *semantic replacement*, where all entities in tweets are *replaced* with their corresponding semantic concepts, the feature space shrunk substantially by nearly 15-20%, and produced an average F measure of 68.9%. However, this accuracy is 3.5% and 10.2% less than when using semantic augmentation and interpolation respectively. The performance degradation is due to the information loss caused by this term replacement, which subsequently reduces NB performance.

Augmenting the original feature space with semantic concepts (*semantic augmentation*) performs slightly better than *sentiment replacement*, though it still performs 6.5% worse than interpolation. With *Semantic interpolation*, semantic concepts are incorporated into NB training taking into account the generative probability of words given concepts. This method produces the highest performance amongst all three incorporation methods, with an average F1-measure of 75.95%.

The contribution of semantic features in the interpolation model is controlled by the interpolation coefficients in Equation 14. We conducted a sensitivity test to evaluate the impact of the interpolation coefficients on sentiment classification accuracy by varying β between 0 and 1. Figure 20 shows that accuracy reaches its peak with β set between 0.3 and 0.5. In our evaluation, we used $\beta = 0.4$ for STS-Expand dataset, and $\beta = 0.3$ for the other two.

Method	STS-Expand	HCR	OMD	Average
Semantic replacement	74.10	61.35	71.25	68.90
Semantic augmentation	77.65	63.65	72.70	71.33
Semantic interpolation	**83.90**	**66.10**	**77.85**	**75.95**

Table 17: Average sentiment classification performance (%) using different methods for incorporating the semantic features. Performance here is the average harmonic mean (F measure) obtained from identifying positive and negative sentiment.

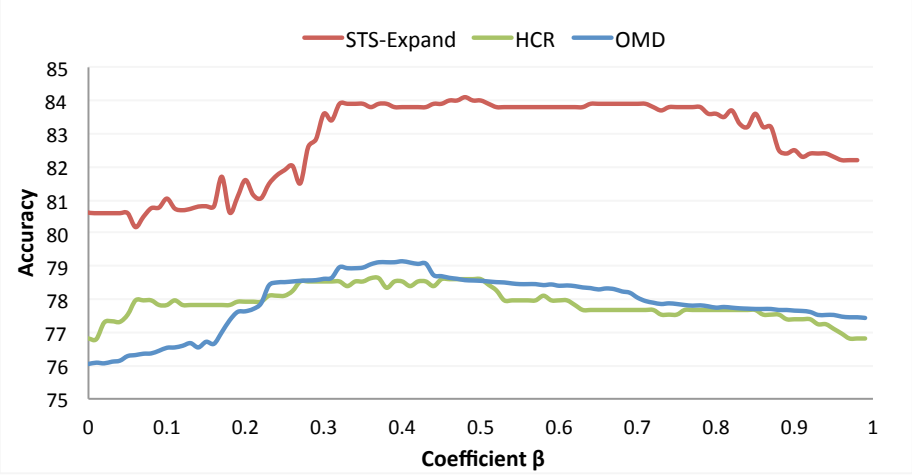


Figure 20: Sensitivity test of the interpolation coefficient for semantic interpolation.

4.2.4.2 *Comparison of Results*

In this section we will compare the Precision, Recall, and F measure of our conceptual semantic sentiment analysis against the baselines described in Section 4.2.3.3. We report the semantic classification results for identifying positive and negative sentiment separately to allow for deeper analysis of results. This is especially important given how some analysis methods perform better in one sentiment polarity than in the other.

Table 18 shows the results of our sentiment classification using *Unigrams*, *POS*, *Sentiment-Topic*, and *Conceptual Semantic* features, applied over the STS-Expand, HCR, and OMD datasets. The table reports three sets of P, R, and F1, one for positive sentiment identification, one for negative sentiment identification, and the third shows the averages of the two.

According to these results in Table 18, the Semantic approach outperforms the Unigrams and POS baselines in all categories and for all three datasets. However, for the HCR and OMD datasets, the sentiment-topic analysis approach seems to outperform the

Dataset	Feature	Positive Sentiment			Negative Sentiment			Average		
		P	R	F1	P	R	F1	P	R	F1
STS-Expand	Unigrams	82.20	75.20	78.50	79.30	85.30	82.20	80.75	80.25	80.35
	POS	83.70	75.00	79.10	79.50	86.90	83.00	81.60	80.95	81.05
	Sentiment-Topic	80.70	82.20	81.40	83.70	82.30	83.00	82.20	82.25	82.20
	Conceptual Semantics	85.80	79.40	82.50	82.70	88.20	85.30	**84.25**	**83.80**	**83.90**
HCR	Unigrams	39.00	36.10	37.50	81.00	82.80	81.90	60.00	59.45	59.70
	POS	56.20	22.00	31.70	80.00	94.70	86.70	68.10	58.35	59.20
	Sentiment-Topic	53.80	47.20	50.30	84.50	87.60	86.00	**69.15**	**67.40**	**68.15**
	Conceptual Semantics	53.60	40.40	46.10	83.10	89.30	86.10	68.35	64.85	66.10
OMD	Unigrams	64.20	70.90	67.10	83.30	78.60	80.80	73.75	74.75	73.95
	POS	69.50	68.30	68.70	83.10	83.90	83.40	76.30	76.10	76.05
	Sentiment-Topic	68.20	75.60	71.70	87.10	82.40	84.70	77.65	**79.00**	**78.20**
	Conceptual Semantics	75.00	66.60	70.30	82.90	88.10	85.40	**78.95**	77.35	77.85

Table 18: Cross comparison results of all the four features.

semantic approach by a small margin. For example, the semantic approach produced higher P, R, and F1 for the STS-Expand dataset, with F1 4.4% higher than Unigrams, 3.5% higher than POS, and 2.1% higher than the sentiment-topic features. In HCR, F1 from the semantic features were 8.9% and 11.7% higher than Unigrams and POS, but 3% lower than F1 from sentiment-topic features. For OMD, semantic features also outperformed the Unigrams and POS baselines, with 5.2% and 2.4% higher F1 respectively. However, in the OMD dataset, F1 from semantic features was 0.4% lower than from the topic model, although Precision was actually higher by 1.7%.

As mentioned in Section 4.2.3.1 and detailed in Table 13, the STS-Expand dataset consists of a large collection of general tweets with no particular topic focus. Unlike STS-Expand, the other two datasets are much smaller in size and their tweets discuss very specific topics; the US Health Care Reform bill in the HCR dataset, and the Obama McCain debate in the OMD dataset. Using conceptual semantic features seem to perform best in the large and general dataset, whereas the sentiment-topic features seem to take the lead in small, topic-focused datasets. The reason is likely to be that classifying with sentiment-topic features group words into a number of topics. In our experiments, we found that for the STS-Expand dataset, increasing the number of topics leads to the increase of classification performance with the peak value of 82.2% in F1-measure reached at topic number 50. Further increasing topic numbers degrades the classifier performance. However, for HCR and OMD, the best performance was obtained with only one topic (F1

= 68.15% for HCR and F1 = 78.20% for OMD). The classification performance drops significantly by any further increment. This could be explained by the nature of these three datasets. HCR was collected using the hashtag "#hcr" (health care reform) while OMD consists of tweets about the Obama-McCain debate. Hence these two datasets are topic-specific. On the contrary, STS was collected using more general queries and thus it contains a potentially large number of topics.

Hence the benefits of using the sentiment-topic features seem to be reduced in comparison to semantic features when the training set is of general content as in the STS tweets dataset.

The average results across all three datasets are shown in Table 19. Here we can see that conceptual semantic features do better than sentiment-topic features and the other baselines when identifying *negative sentiment*. However, sentiment-topic features seem to perform better for *positive sentiment*. For positive sentiment, using the semantic approach produces precision levels that are better than the ones obtained with Unigrams, POS, and sentiment-topic by 15.6%, 2.4%, and 5.8% respectively. However, the Recall produced by the semantic approach when identifying positive sentiment is 2.3% and 12.8% higher than in Unigrams and POS, but 9% lower than Recall from the sentiment-topic approach. Overall, F1 for positive sentiment from semantic features is 2.2% lower than when using sentiment-topic features. It is worth emphasising that the average Precision from identifying both positive and negative sentiment is the highest at 77.18% when using semantic features. When analysing large amounts of continuously flowing data as with social media resources, Precision could well be regarded as much more important than Recall.

Features	**Positive Sentiment**			**Negative Sentiment**			**Average**		
	P	R	F1	P	R	F1	P	R	F1
Unigrams	61.80	60.73	61.03	81.20	82.23	81.63	71.50	71.48	71.33
POS	69.80	55.10	59.83	80.87	88.50	84.37	75.53	72.23	72.48
Sentiment-Topic	67.57	**68.33**	**67.80**	**85.10**	84.10	84.57	77.02	**76.73**	**76.75**
Semantics	**71.47**	62.13	66.30	82.90	**88.53**	**85.60**	**77.18**	75.33	75.95

Table 19: Averages of Precision, Recall, and F measures across all three datasets.

4.3 CONCEPTUAL SEMANTICS FOR LEXICON-BASED SENTIMENT ANALYSIS

So far in this chapter, we presented our approach of extracting and exploiting conceptual semantics as features for sentiment analysis using a supervised machine learning approach.

In this section, we continue investigating with conceptual semantics by using them for lexicon-based sentiment analysis. To this end, we propose incorporating our conceptual semantic features into our previously proposed model; SentiCircles (see Chapter 3). Remember that SentiCircles is a lexicon-based sentiment detection model that uses the contextual semantics of words in order to capture their sentiment in tweets. Therefore, by enriching SentiCircles with conceptual semantics, we aim to: (i) study the usefulness of conceptual semantics when used for lexicon-based sentiment analysis, and (ii) investigate the impact of combining contextual and conceptual semantics of words together on the overall sentiment detection performance of the SentiCircles model.

4.3.1 Enriching SentiCircles with Conceptual Semantics

We add conceptual semantics into the SentiCircle representation using the **Semantic Augmentation** method (Section 4.2.2), i.e., we add the semantic concepts of named-entities to the original tweet before extracting our SentiCircle representation (e.g., "`headache`" and its concept "`Health Condition`" will appear together in the SentiCircle of "`headache`", since by applying semantic augmentation, these terms now co-occur within the augmented tweets). Also note that each extracted concept is represented by a SentiCircle in order to compute its overall sentiment.

The rational behind augmenting semantic concepts into SentiCirlces is that certain entities and concepts tend to have a more consistent correlation to terms of positive or negative sentiment. This can help determine the sentiment of semantically relevant or similar entities which do not explicitly express sentiment. In the example in Figure 21, "`Wind`" and "`Humidity`" have negative SentiCircles as they tend to appear with negative terms in tweets. Hence their concept "`Weather Condition`" will have a stronger negative sentiment. The tweet "`Cycling under a heavy rain.. What a #luck!`" is likely to have a negative sentiment due to the presence of the word "`rain`" which is mapped to the negative concept "`Weather Condition`". Moreover, the word `heavy` in this context is

more likely to have a negative sentiment due to its correlation with "`rain`" and "`Weather Condition`".

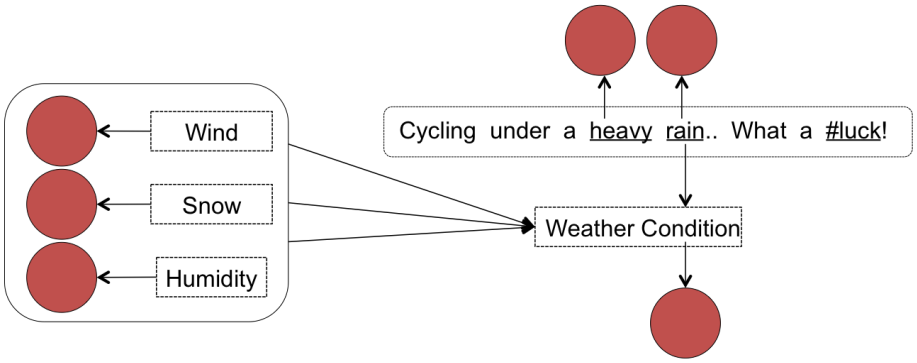


Figure 21: Mapping semantic concepts to detect sentiment.

4.3.2 Evaluation Results

Here, we report the results when enriching the SentiCircle representation with conceptual semantics by using the augmentation method. In particular, we perform binary sentiment detection at the tweet-level using the three SentiCircle detection methods, Median, Pivot-Method, and Pivot-Hybrid on the OMD, HCR and STS-Gold datasets (see Section 3.3.2).

The number of entities and concepts extracted for HCR and OMD is the one reported in Table 14. For the STS-Gold dataset, 2735 entities and 23 distinct concepts were extracted using AlchemyAPI.

We perform 10-fold cross validation on all datasets and report the win/loss in accuracy and F-measure when adding the conceptual semantics to the SentiCircle model with respect to the original results reported in Chapter 3 (Figure 15) when applying SentiCircles without semantic augmentation. Note that here we used Thelwall-Lexicon [160] to obtain the word prior sentiments in our three sentiment detection methods. As mentioned in Chapters 2 and 3, Thelwall-Lexicon is designed to specifically work on social data.

The results, as depicted in Figure 22, show that the impact of conceptual semantics on the performance of SentiCircles varies across datasets. On the HCR dataset, a 1.21% gain in F-measure is obtained using the Median method. On the STS-Gold dataset, the Median and the Pivot methods improve the performance by 0.49% and 0.12% in F-measure, respectively. On the other hand, a small drop in both accuracy and F-measure is observed on the OMD dataset when using conceptual semantics with either method.

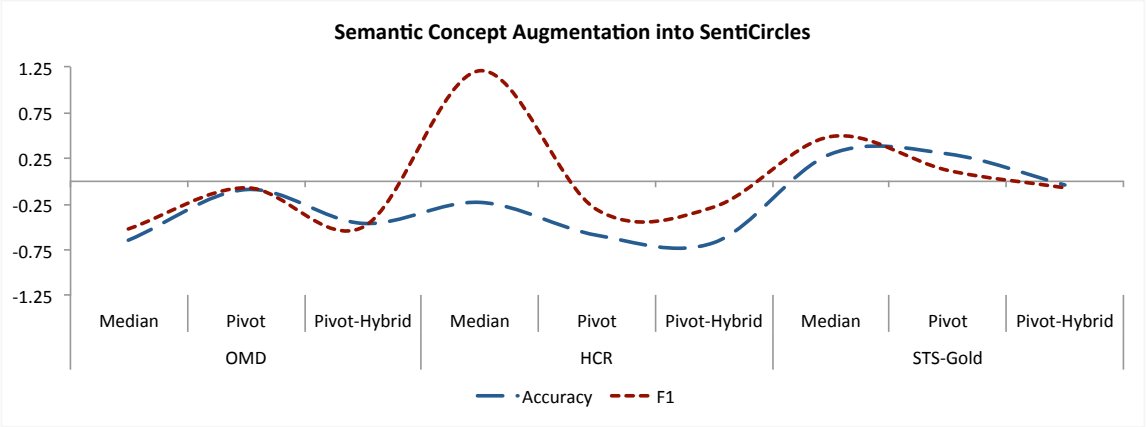


Figure 22: Average Win/Loss in Accuracy and F-measure of incorporating conceptual semantics into SentiCircles using the Median, Pivot and Pivot-Hybrid methods on all datasets.

It is also worth noting that the Median method is more affected by conceptual semantic incorporation than the Pivot method, where a much clearer shift in performance can be observed across datasets. This can be explained as the Median method considers all the incorporated concepts in the SentiCircle, whereas the Pivot method focus more on concepts that are associated to target terms in tweets (Section 3.3.2).

Overall, the average results across the three datasets show that, using conceptual semantics with the Median method, always improve the sentiment detection performance by 0.39%, on average. On the other hand, both the Pivot and Pivot-Hybrid methods lower the overall performance by 0.09% and 0.28% in F-measure respectively. Also, it seems that conceptual semantics with the Pivot-Hybrid method, does not bring much improvement to the performance on either dataset.

Lastly, the number of entities extracted for the STS-Gold dataset is almost twice as for HCR or OMD. Nonetheless, the average results show that semantic incorporation seems to have a better impact on the STS-Gold dataset than on the OMD and HCR datasets. This might be due to the topical-focus of each dataset. As discussed earlier, the OMD and HCR are both composed of a smaller number of tweets about specific topics (the US Health Care Reform bill and the Obama-McCain debate), the STS-Gold dataset contains a larger number of tweets with no particular topical focus.

4.4 DISCUSSION

In this chapter we demonstrated the value of using conceptual semantics as features for the classification of positive and negative sentiment in Tweets. We aimed at addressing

the question of whether incorporating conceptual semantics into supervised and lexicon-based methods increases their sentiment classification performance or not.

Our methodology of using conceptual semantics for sentiment analysis consists mainly of two steps: conceptual semantics extraction and conceptual semantics incorporation.

Conceptual Semantic Extraction: We tested several off-the-shelf semantic entity extractors and decided on using AlchemyAPI due to its better performance in terms of coverage and accuracy. However, in our evaluation of AlchemyAPI, Zemanta, and OpenCalais, we observed that some of them perform better than others for specific type of entities. For example, Zemanta produced more accurate and specific concepts to describe entities related to music tracks and bands. It might be possible to implement a more selective approach, where certain semantic extractors and concept identifiers are used, or trusted more, for certain type of entities.

Conceptual Semantic Incorporation: We proposed adding semantic concepts as features into: (i) the training phase of NB supervised classifier and (ii) the semantic representation phase of our SentiCircle lexicon-based model. For both models all semantic concepts extracted from the tweets were added to the analysis. However, it might be the case that semantic features improve sentiment analysis accuracy for some types of concepts (e.g. cities, music) but reduce accuracy for some other concept types (e.g. people, companies). Therefore, a potential future direction of this work is to investigate the impact of each group of concepts on our analysis accuracy, to determine their individual contribution and impact on sentiment analysis. We can also assign weights to each concept type to represent its correlation with positive or negative sentiment.

We experimented with multiple datasets of varying sizes and topical-focus. Our results showed that accuracy can be sensitive to the size of the datasets and their topical-focus. For example, our evaluation, on both supervised and lexicon-based settings, showed that the semantic approach excels when the dataset is large and of diverse topic coverage. An area for a future work will be to apply these approaches on larger datasets to examine the consistency of their performance patterns. Another possible future work is to explore various feature selection strategies to improve the sentiment classification performance.

4.5 SUMMARY

Conceptual semantics refer to those semantically hidden concepts of named entities extracted from tweets. In this chapter, we proposed the use of conceptual semantics in Twitter sentiment analysis. We aimed at addressing the research question: *Could the conceptual semantics of words enhance sentiment analysis performance?*. To this end, we proposed using semantic concepts as features into two widely used sentiment analysis approaches on the Twitter sentiment analysis literature; the supervised machine learning approach and the lexicon-based approach.

For supervised sentiment classification, we explored three different methods for incorporating conceptual semantics into the analysis; with replacement, augmentation, and interpolation. We found that best results were achieved when interpolating the generative model of words given semantic concepts into the unigram language model of the NB classifier. We experimented with three Twitter datasets and compared the semantic features with the the Unigrams and POS features as well as with the sentiment-topic features. Our results show that the semantic feature model outperforms the Unigram and POS baseline for identifying both negative and positive sentiment. We demonstrated that adding semantic features produces higher Recall and F1 score, but lower Precision, than sentiment-topic features when classifying negative sentiment. We also showed that using semantic features outperforms the sentiment-topic features for positive sentiment classification in terms of Precision, but not in terms of Recall and F1. One average, the semantic features appeared to be the most precise amongst the four other feature selections we experimented with.

For lexicon-based sentiment analysis, we enriched our SentiCircle representation with conceptual semantics using the augmentation method. Results on three Twitter datasets showed that adding concepts to SentiCircle has a good potential over using SentiCircles solely.

Overall, our results suggest that the semantic approach is more appropriate when the datasets being analysed are large and cover a wide range of topics, whereas the sentiment-topic approach was most suitable for relatively small datasets with specific topical foci.

5

SEMANTIC PATTERNS FOR SENTIMENT ANALYSIS OF TWITTER

In previous chapters we explored the value of the contextual and conceptual semantics of words in sentiment analysis on Twitter. In this chapter, we move a step further; instead of using semantics associated with individual words as indicators to sentiment, we propose extracting patterns from the contextual semantics and sentiment similarities between words in tweets. After that, we investigate how the extracted patterns can be used to enhance sentiment classification performance on Twitter.

Our findings suggest that contextual semantic and sentiment similarities of words do exist in tweets, by means of certain patterns. These patterns, when used as features to train sentiment classifiers, consistently outperform other types of features extracted from the syntactic, contextual or conceptual semantic representations of words.

5.1 INTRODUCTION

Chapters 3 and 4 showed that merely relying on the individual words does not always lead to satisfactory sentiment detection results since sentiment is often context-dependent. In this chapter we propose extracting and using the semantics of *individual words* to detect their sentiment as well as the overall sentiment of the tweet they occur within.

By doing this research, we observed that some words had similar contextual semantics in different tweets (i.e., similar co-occurrence patterns and contextual term distributions). For example, similar contextual semantics are likely to be used when expressing positive sentiment towards "`Beatles`" and "`Katy Perry`". On the other hand, the contextual semantics for "`iPod`" and "`Taylor Swift`" are likely to be different even though they both receive positive sentiment as discussed in Chapter 3. Such semantic similarities between words can be used to form clusters that represent different patterns of semantics and sentiment. These patterns, when identified and extracted from tweets, can be used to enhance the overall sentiment classification performance of tweets as well as named entities. Knowing the semantic pattern that is used to express a certain sentiment enables

the identification of the sentiment of other words or entities following the same pattern. We refer to these patterns as semantic sentiment patters or shortly as *SS-Patterns*.

The research question we aim to address in this chapter is:

RQ3 ***Could semantic sentiment patterns boost sentiment analysis performance?***

To address the above question, we propose a new approach for automatically extracting patterns based on the contextual semantic and sentiment similarities between words in a given Twitter corpus.

We apply our approach to 9 different Twitter datasets, and validate the extracted patterns by using them as classification features in entity- and tweet-level sentiment analysis tasks. To this end, we train several supervised classifiers from SS-Patterns and compare the sentiment classification performance against models trained from 6 state-of-the-art sets of features derived from both the syntactic and semantic representations of words.

Our results show that SS-Patterns consistently outperform all the baseline feature sets, on all 9 datasets, in both tweet-level and entity-level sentiment classification tasks. At the tweet level, SS-Patterns improve the classification performance by 1.94% in accuracy and 2.19% in F-measure on average. Also, at the entity level, our patterns produce 2.88% and 2.64% higher accuracy and F-measure than all other features respectively.

We also conduct quantitative and qualitative analyses on a sample of the patterns extracted by our approach and show that the effectiveness of using SS-Patterns as additional features for classifier training is attributed to their ability to capture words with similar contextual semantics and sentiment. We also show that our extraction approach is able to detect patterns of controversial sentiment (strong opposing sentiment) expressed by people towards certain entities.

The rest of this chapter is organised as follows: Background about using linguistic and semantic patterns in sentiment analysis is presented in Section 5.2. The proposed approach for extracting semantic sentiment patterns is presented in Section 5.3. Evaluation setup and results are presented in Sections 5.4 and 5.5 respectively. Our pattern analysis study is described in Section 5.6. Discussion is covered in Section 5.7. Finally, we summarise our work and findings in this chapter in Section 5.8.

5.2 RELATED WORK

Sentiment in text is often conveyed through relations and dependencies between words, which often formulate sentiment [130]. These relations are usually compiled as a set of syntactic patterns (i.e., Part-of-Speech patterns) [165, 130, 161] or semantic and common sense concepts [55, 25]. For example, the adjective word "`mean`" when preceded by a verb, constitutes a pattern of negative sentiment as in: "`she said mean things`". Also, the word "`destroy`" formulates a positive pattern when occurs with the concept "`invading germs`". However, both syntactic and semantic approaches to extracting sentiment patterns are usually not tailored to Twitter due to its noisy nature. Also they both function with external knowledge sources. Most syntactic approaches rely on fixed and pre-defined sets of syntactic templates for pattern extraction. On the other hand, semantic approaches rely on external ontologies and common sense knowledge bases. Such resources, although useful, tend to be focused on particular domains, and therefore provide limited coverage, which is especially problematic when processing general Twitter streams, with their rapid semiotic evolution and language deformations, as discussed in Chapter 4.

In order to overcome the aforementioned limitations and to address the third research question of this thesis, we design our sentiment pattern extraction approach in a way that captures patterns based on the contextual semantic and sentiment similarities between words in a Twitter corpus. Our approach does not rely on the syntactic structures in tweets, nor requires using pre-defined syntactic template sets or external semantic knowledge sources.

5.3 SEMANTIC SENTIMENT PATTERNS OF WORDS

We define semantic sentiment patterns as clusters of words which have similar contextual semantics and sentiment in text. Based on this definition, the problem of capturing these patterns in tweets data breaks down into three phases as illustrated in Figure 23. In the first phase, tweets in a given data collection are syntactically processed in order to reduce the amount of noise and language informality in them (Section 5.3.1). In the second phase we apply our SentiCircle representation model (see Chapter 3) on the processed tweets to capture the contextual semantics and sentiment of words (Section 5.3.2). In the third

phase the semantic sentiment patterns are formed by clustering words that share similar semantics and sentiment (i.e., similar SentiCircles) (Section 5.3.3).

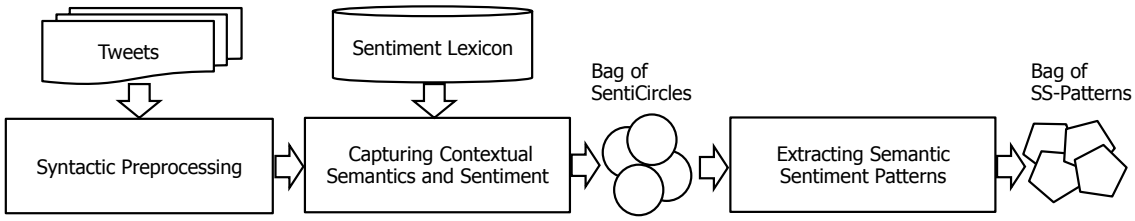


Figure 23: The systematic workflow of capturing semantic sentiment patterns from Twitter data.

In the subsequent sections we describe each of the aforementioned phases in more detail:

5.3.1 Syntactical Preprocessing

Tweets are usually composed of incomplete, noisy and poorly structured sentences due to the frequent presence of abbreviations, irregular expressions, ill-formed words and non-dictionary terms. Such noisy nature of tweets has been shown to indirectly affect the sentiment classification performance [137]. Therefore, this phase applies the following pre-processing steps to reduce the amount of noise in the tweets and consequently reduce its possible negative impact on the extraction of SS-Patterns:

- Replace all URL links in the Twitter corpus with the term "`URL`".
- Remove all non-ASCII and non-English characters.
- Revert words that contain repeated letters to their original English form. For example, the word "`maaadddd`" will be converted to "`mad`" after processing.
- Process contraction and possessive forms. For example, change "`he's`" and "`friend's`" to "`he`" and "`friend`".

Note that we do not remove stopwords from the data since they tend to carry sentiment information as will be shown in Chapter 6.

5.3.2 Capturing Contextual Semantics and Sentiment of Words

SS-patterns are formed from the contextual semantic similarities among words. Therefore, a key step in our pipeline is to capture the words' contextual semantics in tweets. To this end, we use our semantic representation model, SentiCircle (Chapter 3).

Remember that the SentiCircle model extracts the contextual semantics of a word from (i) its co-occurrences with other words in a given tweet corpus and (ii) its prior sentiment obtained from an external sentiment lexicon. Therefore, this phase takes as input a collection of preprocessed tweets and a sentiment lexicon, and returns a bag of SentiCircles that represent each unique word in the tweet collection.

Also, remember that the overall contextual semantics and sentiment of a given word can be computed by extracting the SentiMedian of the word's SentiCircle, as described in Section 3.2.2.4.

5.3.3 Extracting Patterns from SentiCircles

At this stage all the unique words in the tweet collection have their contextual semantics and sentiment extracted and represented by means of their SentiCircles. It is very likely to find words in text that share similar contextual semantics and sentiment. In other words, finding words with similar SentiCircles. Therefore, this phase seeks to find such potential semantic similarities in tweets by building clusters of similar SentiCircles. The output of this phase is a set of clusters of words, which we refer to as the *semantic sentiment patterns of words (SS-Patterns)*.

SentiCircles Clustering: We can capture patterns that emerge from the similarity of word's sentiment and contextual semantics by clustering the SentiCircles of those words. In particular, we perform a clustering task fed by dimensions that are provided by SentiCircles; *density*, *dispersion*, and *geometry*. Density and dispersion usually characterise terms and entities that receive controversial sentiment in tweets as will be further explained and validated in Section 5.6. Geometry, on the other hand, preserves the contextual sentiment orientation and strength of terms. Once we extract the vectors that represent these three dimensions for all the terms' SentiCircles, we feed them into a common clustering method; k-means.

In the following, we describe the three dimensions we extract from each term's SentiCircle Ω along with the components they consist of:

- *Geometry*: includes the X- and Y-component of the SentiMedian $g(x_g, y_g) \in \Omega$
- *Density*: includes the total density of points in the SentiCircle Ω and its computed as: $density(\Omega) = N/M$, where N is the total number of points in the SentiCircle and M is the total number of points in the SentiCircles of all terms.

We also compute five density components, representing the density of each sentiment quadrant in the SentiCircle (i.e., positive, very positive, negative and very negative quadrants) along with the density of its neutral region. Each of these components is computed as $density(Q) = P/N$ where P is the total number of points in the sentiment quadrant Q.

- *Dispersion*: the total dispersion of a SentiCircle refers to how scattered, or condensed, the points (context terms) in the circle are. To calculate the value of this component we use the *median absolute deviation measure (MAD)* [169], which computes the dispersion of Ω as the median of the absolute deviations from the SentiCircle's median point (i.e., the SentiMedian g_m) as:

$$mad(\Omega) = (\sum_{i=1}^{n} |p_i - g_m|)/N \tag{16}$$

where p_i represents the coordinates of the point i within the SentiCircle, and N is the total number of points within the SentiCircle.

Similarly, using the above equation, we calculate the dispersion of each sentiment quadrant and the neutral region in the SentiCircle. We also calculate the dispersion of the active region in SentiCircle (i.e., The SentiCircle after excluding points in the neutral region)

The last step in our pipeline is to apply k-means on all SentiCircles (represented by their associated vectors capturing the above three mentioned dimensions). This results in a set of clusters $\mathcal{K} = (k_1, k_2, ..., k_c)$ where each cluster consists of words that have similar contextual semantics and sentiment. We call $\mathcal{K}$ the set of clusters (or patterns) and $k_i \in \mathcal{K}$ the *semantic sentiment pattern*.

In the subsequent section, we describe how to determine the number of patterns (clusters) in the data and how to validate the extracted patterns by using them as features in two sentiment classification tasks.

5.4 EVALUATION SETUP

Our proposed approach, as shown in the previous section, extracts patterns of words of similar contextual semantics and sentiment. We evaluate the extracted SS-patterns by using them as classification features to train supervised classifiers for two sentiment analysis tasks, tweet- and entity-level sentiment classification. To this end, we use 9 publicly

and widely used datasets in Twitter sentiment analysis literature [138]. Nine of them will be used for tweet-level evaluation and one for entity-level evaluation. As for evaluation baselines, we use 6 types of classification features and compare the performance of classifiers trained from our SS-patterns against those trained from these baseline features.

5.4.1 Tweet-Level Evaluation Setup

The first validation test we conduct on our SS-patterns is to measure their effectiveness as features for binary sentiment analysis of tweets, i.e., classifying the individual tweets as positive or negative. To this end, we use SS-patterns extracted from a given Twitter dataset to train two supervised classifiers popularly used for tweet-level sentiment analysis, Maximum Entropy (MaxEnt) and Naïve Bayes (NB) from Mallet.[1] We use 9 different Twitter datasets in our validation in order to avoid any bias that a single dataset can introduce. Numbers of positive and negative tweets within these datasets are summarised in Table 20, and detailed in Appendix A.

Dataset	Tweets	#Negative	#Positive	#Unigrams
Stanford Twitter Test Set (STS-Test) [60]	359	177	182	1562
Sanders Dataset (Sanders) [138]	1224	654	570	3201
Obama McCain Debate (OMD) [47]	1906	1196	710	3964
Health Care Reform (HCR) [152]	1922	1381	541	5140
Stanford Gold Standard (STS-Gold) [138]	2034	632	1402	4694
Sentiment Strength Twitter Dataset (SSTD) [160]	2289	1037	1252	6849
The Dialogue Earth Weather Dataset (WAB) [5]	5495	2580	2915	7485
The Dialogue Earth Gas Prices Dataset (GASP) [5]	6285	5235	1050	8128
Semeval Dataset (Semeval) [104]	7535	2186	5349	15851

Table 20: Twitter datasets used for tweet-level sentiment analysis evaluation.

5.4.2 Entity-Level Evaluation Setup

In the second validation test, we evaluate the usefulness of SS-Patterns as features for entity-level sentiment analysis, i.e., detecting sentiment towards a particular entity. To this end, we perform a 3-way sentiment classification (negative, positive, neutral) on a dataset of 58 named entities extracted from the STS-Gold dataset [138] and manually

1 http://mallet.cs.umass.edu/

labelled with their sentiment class. Numbers of negative, positive and neutral entities in this dataset are listed in Table 21 along with five examples of entities under each sentiment class. Details of the extraction and the annotation of these entities can be found in Chapter 3.

	Negative Entities	**Positive Entities**	**Neutral Entities**
Total Number	13	29	16
Examples	Cancer	Lakers	Obama
	Lebron James	Katy Perry	Sydney
	Flu	Omaha	iPhone
	Wii	Taylor Swift	Youtube
	Dominique Wilkins	Jasmine Tea	Vegas

Table 21: Numbers of negative, positive and neutral entities in the STS-Gold Entity dataset along with examples of 5 entities under each sentiment class.

The entity sentiment classifier we use in our evaluation is based on maximum likelihood estimation (MLE). Specifically, we use tweets in the STS-Gold dataset to estimate the conditional probability $P(c|e)$ of an entity e assigned with a sentiment class $c \in \{Positive, Negative\}$ as:

$$P(c|e) = N(e,c)/N(e) \tag{17}$$

where $N(e,c)$ is the frequency of an entity e in tweets assigned with a sentiment class c and $N(e)$ is the frequency of the entity e in the whole corpus.

We incorporate our SS-Pattern features and other baseline features (Section 5.4.3) into the sentiment class estimation of e by using the following back-off strategy:

$$\hat{c} = \begin{cases} P(c|e) & \text{if } N(e,c) \neq 0 \\ P(c|f) & \text{if } N(e,c) = 0 \end{cases} \tag{18}$$

where f is the incorporated feature (e.g., the SS-Pattern of e) and $P(c|f)$ is the conditional probability of the feature f assigned with a sentiment class c and it can be also estimated using MLE. The rationale behind the above back-off strategy is that some entities might not occur in tweets of certain sentiment class, leading therefore, to zero probabilities. In such cases we resort to the sentiment of the latent features associated with these entities in the dataset.

To extract the sentiment of e we first calculate the following ratio:

$$R_e = P(c = Positive|e)/P(c = Negative|e) \tag{19}$$

In particular, the sentiment is neutral if R_e is close enough to 1, i.e., $R_e \in [1-\lambda, 1+\delta]$, otherwise the sentiment is negative if $R_e < 1-\lambda$ or positive if $R_e > 1+\delta$. We determine the value of λ and δ by plotting the ratio R_e for all the 58 entities and check the upper and lower cut-offs between which the plot converges. In our case, $\lambda \approx 0.7$ and $delta \approx 0.3$, as shown in Figure 24.

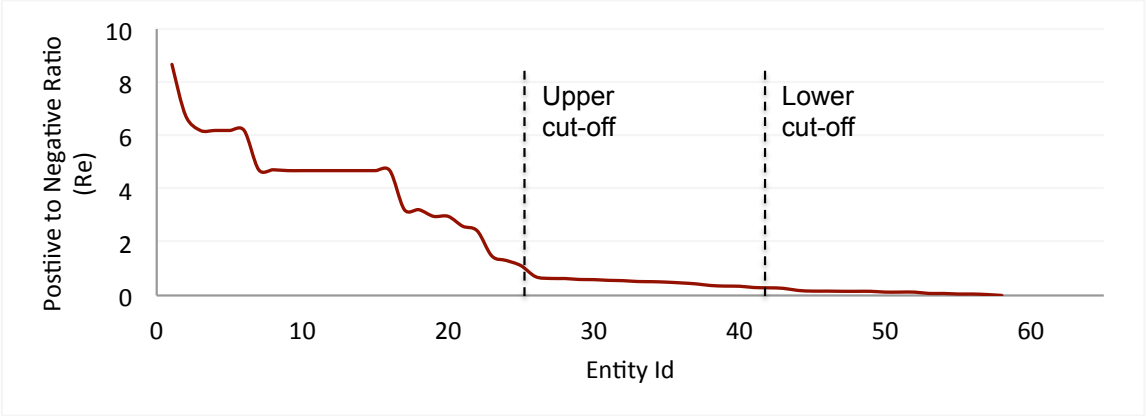


Figure 24: Positive to negative sentiment ratio for each of the 58 entities in the STS-Gold dataset.

Note that in the case where $P(c = Negative|e)$ or $P(c = Negative|e)$ is zero when computing R_e (Equation 19), we resort to using the conditional probability of the entity-associated feature with the given sentiment class c (i.e., $P(c = Negative|f)$ or $P(c = Positive|f)$).

5.4.3 Evaluation Baselines

The baseline model in our evaluation is a sentiment classifier trained from word unigram features. Table 20 shows the number of unique unigram features extracted from our datasets.

In addition to unigrams, we propose comparing our SS-Pattern features against the below described five state-of-the-art types of features in sentiment analysis. Amongst them, two sets of features are derived from the syntactical characteristics of words in tweets (POS features, and Twitter features), one is based on the prior sentiment orientation of words (Lexicon features) and two are obtained from the semantic representation of words in tweets (Semantic Concept features and LDA-Topic features):

Twitter Features: refer to tokens and characters that are popularly used in tweet messages such as hashtags (e.g., "`#smartphone`"), user mentions (e.g, "`@obama`"), the tweet reply token ("`RT`") and emoticons (e.g., "`:) :D <3 o_O`").

Part-of-Speech Features: refer to the part-of-speech tags of words in tweets (e.g., verbs, adjectives, adverbs, etc). We extract these features using the TweetNLP POS tagger.[2]

Lexicon Features: these features are formed from the opinionated words in tweets along with their prior sentiment labels (e.g., "`excellent_positive`", "`bad_negative`", "`good_positive`", etc.). We assign words with their prior sentiments using both Thelwall [160] and MPQA [176] sentiment lexicons.

Semantic Concept Features: refer to our conceptual semantic features presented in Chapter 4, i.e., the semantic concepts (e.g., "`person`", "`company`", "`city`") that represent entities (e.g., "`Obama`", "`Motorola`", "`Vegas`") appearing in tweets. To extract the entities and their associated concepts in our datasets we use AlchemyAPI.[3] An assessment of the semantic extraction performance of AlchemyAPI on Twitter data is presented in Chapter 4. The number of extracted concepts in each dataset is listed in Table 22.

Dataset	No. of Semantic Concepts
STS-Test	299
Sanders	1407
OMD	2191
HCR	1626
STS-Gold	1490
SSTD	699
WAB	1497
GASP	3614
Semeval	2875

Table 22: Numbers of the semantic concepts extracted from all datasets.

LDA-Topic Features: these features denote the latent topics extracted from tweets using the probabilistic generative model, LDA [19]. LDA assumes that a document is a mixture of topics and each topic is a mixture of probabilities of words that are more likely to

2 http://www.ark.cs.cmu.edu/TweetNLP/

3 http://www.alchemyapi.com

co-occur together under the topic (see Chapter 2). For example the topic "`iPhone`" is likely to be associated with words like "`display`" and "`battery`". Therefore, LDA-Topics represent groups of words that are semantically related. To extract these latent topics from our datasets we use an implementation of LDA provided by Mallet. LDA requires defining the number of topics to extract before applying it on the data. To this end, we ran LDA with different number of topics (e.g., 1 topic, 10 topics, 20 topics, 30 topics, etc). Among all choices, 10 topics was the number that gave the highest sentiment classification performance when the topics were incorporated into the feature space.

Note that all the above sets of features are combined with the original unigram features when training the baseline sentiment classifiers for both entity- and tweet-levels.

5.4.4 Number of SS-Patterns in Data

As described earlier, extracting SS-patterns is a clustering problem that requires determining beforehand the number of clusters (patterns) to extract. To this end, we run k-means multiple times with k varying between 1 and 100. We then plot the within-cluster sum of squares for all the outputs generated by k-means. The optimum number of clusters is found where an "*elbow*" appears in the plot [99]. For example, Figure 25 shows that the optimum number of clusters for the GASP dataset is 17, which in other words, represents the number of SS-Patterns features that our sentiment classifiers should be trained from. Table 23 shows the number of SS-Patterns extracted by our model for each dataset.

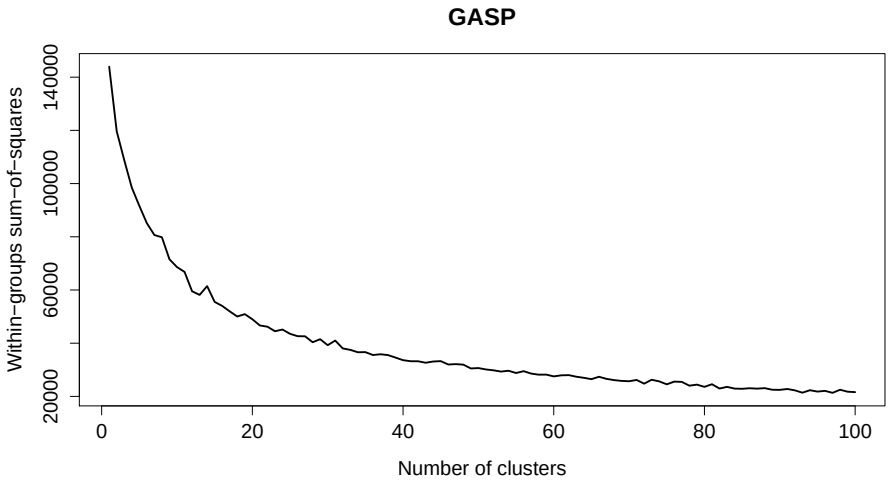


Figure 25: Within-cluster sum of squares for different numbers of clusters (SS-Patterns) in the GASP dataset.

Dataset	No. of SS-Patterns
STS-Test	18
Sanders	20
OMD	23
HCR	22
STS-Gold	26
SSTD	24
WAB	17
GASP	17
Semeval	19

Table 23: Numbers of SS-Patterns extracted from all datasets.

5.5 EVALUATION RESULTS

In this section, we report the results from using our proposed SS-Patterns as features for tweet- and entity-level sentiment classification tasks and compare them against the baselines described in Section 5.4.3. All experiments in both evaluation tasks are done using 10-fold cross validation.

5.5.1 Results of Tweet-Level Sentiment Classification

The first task in our evaluation aims to asses the usefulness of SS-Patterns as features for binary sentiment classification of tweets (positive vs. negative).[4] We use NB and MaxEnt classifiers trained from word unigrams as the starting baseline models (unigram models). We then compare the performance of classifiers trained from other types of features against these unigram models.

Table 24 shows the results in accuracy and average F1 measure of both unigram models across all datasets. The highest accuracy, 90.49%, is achieved on the GASP dataset using MaxEnt, while the highest average F-measure, 84.08%, is obtained on the WAB dataset. On the other hand, the lowest performance in accuracy, 72.36%, is obtained using NB

4 Unlike entity-level, we do not perform 3-way classification (positive, negative, netrual) in this task since not all the 9 datasets contain tweets of neutral sentiment.

on the SSTD dataset. Also, NB produces the lowest F1, 66.69%, on the HCR dataset. On average, MaxEnt outperforms NB by 1.04% and 1.35% in accuracy and F1 respectively. Hence, we use MaxEnt only to continue our evaluation in this task.

	Dataset	STS-Test	Sanders	OMD	HCR	STS-Gold	SSTD	WAB	GASP	SemEval	**Average**
MaxEnt	Acc	77.82	83.62	82.90	77.02	86.02	72.84	84.12	90.49	82.11	**81.88**
	F1	77.94	83.58	81.34	69.10	83.10	72.27	84.08	81.81	77.03	**78.91**
NB	Acc	81.06	82.66	81.57	74.27	84.22	72.36	82.79	88.16	80.44	**80.84**
	F1	81.07	82.52	79.93	66.69	80.46	72.20	82.74	78.15	74.35	**77.57**

Table 24: Average classification accuracy (acc) and average harmonic mean measure (F1) obtained from identifying positive and negative sentiment using unigram features.

Table 25 shows the results of MaxEnt classifiers trained from the 5 baseline sets of features (see Section 5.4.3) as well as MaxtEnt trained from our proposed SS-patterns, applied over all datasets. The table reports the average results in three sets of *minimum*, *maximum*, and *average* win/loss in accuracy and F-measure relating to the results of the unigram model in Table 24. For simplicity, we refer to MaxEnt classifiers trained from any syntactic feature set as *syntactic models* and we refer to those trained from any semantic feature set as *semantic models*.

It can be observed from these results in Table 25 that all syntactic and semantic models outperform on average the unigram model in both accuracy and F-measure. However, MaxEnt trained from our SS-Patterns significantly outperforms those models trained from any other set of features. In particular, our SS-Patterns produce on average 3.05% and 3.76 % higher accuracy and F1 than the unigram model. This is 2% higher performance than the average performance gain of all syntactic and semantic models. Moreover, we get a maximum improvement in accuracy and F-measure of 9.87% and 9.78% respectively over the unigram model when using our SS-Patterns for training. This is at least 3.54% and 3.61% higher than any other model. It is also worth noting that on the GASP dataset, where the minimum performance gain is obtained, MaxEnt trained from SS-Patterns gives a minimum improvement of 0.70%, while all other models suffer an average performance loss of -0.45%.

Finally, we notice that syntactic features, and more specifically the lexicon ones, are highly competitive in comparison to the semantic features. For example, lexicon features slightly outperform concept and LDA-Topic features. However, from the average performance in Table 25 of both, syntactic and semantic types of features, we can see that

	Features	MaxEnt Classifier					
		Accuracy			F1		
		Minimum	Maximum	Average	Minimum	Maximum	Average
Syntactic	Twitter Features	-0.23	3.91	1.24	-0.25	4.53	1.62
	POS	-0.89	2.92	0.79	-0.91	5.67	1.25
	Lexicon	-0.44	4.23	1.30	-0.38	5.81	1.83
	Average-Syntactic	-0.52	3.69	1.11	-0.52	5.33	1.57
Semantic	Concepts	-0.22	2.76	1.20	-0.40	4.80	1.51
	LDA-Topics	-0.47	3.37	1.20	-0.68	6.05	1.68
	SS-Patterns	**0.70**	**9.87**	**3.05**	**1.23**	**9.78**	**3.76**
	Average-Semantic	0.00	5.33	1.82	0.05	6.88	2.32

Table 25: Win/Loss in Accuracy and F1 of using different features for sentiment classification on all nine datasets.

semantic models are still surpassing syntactic models in both accuracy and F-measure by 0.71% and 0.75% on average respectively.

5.5.2 Results of Entity-Level Sentiment Classification

In this section, we report the evaluation results of using our SS-Patterns for entity-level sentiment classification on the STS-Gold Entity dataset using the entity sentiment classifier described in Section 5.4.2. Note that STS-Gold is the only dataset among the other 9 that provides named entities manually annotated with their sentiment labels (positive, negative, neutral). Therefore, our evaluation in this task is done using the STS-Gold dataset only.

Features	Accuracy	Positive Sentiment			Negative Sentiment			Neutral Sentiment			Average		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Unigrams	74.14	88.46	79.31	83.64	63.16	92.31	75	61.54	50	55.17	71.05	73.87	71.27
LDA-Topics	74.14	100	72.41	84	66.67	76.92	71.43	54.55	75	63.16	73.74	74.78	72.86
Semantic Concepts	75.86	80	82.76	81.36	75	69.23	72	68.75	68.75	68.75	74.58	73.58	74.04
SS-Patterns	**77.59**	100	79.31	88.46	61.11	84.62	70.97	64.71	68.75	66.67	**75.27**	**77.56**	**75.37**

Table 26: Accuracy and averages of Precision, Recall, and F1 measures of entity-level sentiment classification using different features.

Table 26 reports the results in accuracy, precision (P), recall (R) and F1 of positive, negative and neutral sentiment classification performances from using unigrams, semantic

concepts, LDA-Topics and SS-Patterns features. Generally, our SS-Patterns outperform all other features including word unigrams in all measures, on average. In particular, merely using word unigrams for classification gives the lowest performance, 74.14% in accuracy and 71.27% in average F1. However, augmenting the feature space with SS-Patterns improves the performance by 3.45% in accuracy and 4.1% in average F1. Our SS-Patterns also outperform LDA-Topics and semantic concepts features by at least 1.73% and 1.33% in accuracy and average F1 respectively.

As for per-class sentiment classification, we observe that all features produce high performance on detecting positive entities in comparison with detecting negative or neutral ones. For example, the average F1 for detecting negative entities is 72.35%, which is 12% lower than the average F1 for detecting positive entities. Also, we notice that the performance for detecting neutral entities is the lowest with 63.43% in F1, which is 8.9% and 20.9% lower than F1 for detecting negative and positive entities respectively.

Such varying performance might be due to the uneven sentiment class distribution in the entity dataset. As can be noted from Table 21, positive entities constitute 50% of the total number of entities while the neutral and negative entities form together the other 50%.

5.6 WITHIN-PATTERN SENTIMENT CONSISTENCY

Our approach, by definition, seeks to find SS-Patterns of terms of similar contextual semantics and sentiment. Therefore, SS-Patterns are more informative when they are consistent with the sentiment of their terms, that is, they consist mostly of terms of similar contextual sentiment orientations. In this section, we further study the sentiment consistency of our patterns on a set of 14 SS-Patterns extracted from the 58 annotated entities in the STS-Gold dataset. The number of patterns was determined based on the elbow method, as explained in Section 5.4.4.

Table 27 shows four of the extracted patterns along with their top 5 associated entities and the true sentiment of these entities, available within the STSGold dataset. Patterns 3, 12 and 11 are strongly consistent since all entities within them have the same sentiment. On the other hand, Pattern 5 has low sentiment consistency as it contains entities of mixed sentiment orientations.

We systematically calculate the sentiment consistency of a given SS-Pattern k_i as:

$$consistency(k_i) = \arg\max_{s \in \mathcal{S}} \frac{E_s}{E'} \tag{20}$$

where $s \in \mathcal{S} = \{Positive, Negative, Neutral\}$ is the sentiment label, E_s is the number of entities of sentiment s and E' is the total number of entities within K_i.

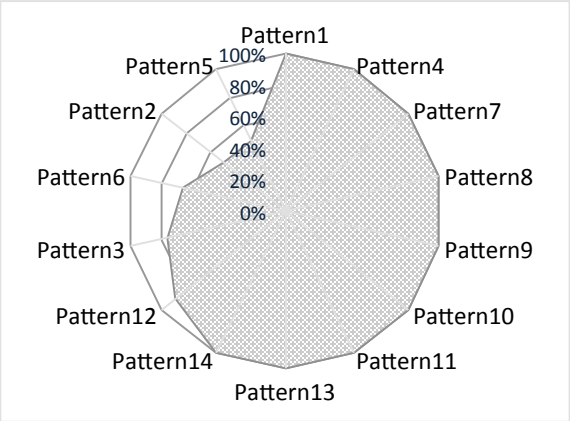


Figure 26: Within-Cluster sentiment consistencies in the STS-Gold Entity dataset.

Figure 26 depicts the sentiment consistency of the 14 SS-Patterns. 9 patterns out of 14 are perfectly consistent with the sentiment of their entities while two patterns have a consistency higher than 77%. Only patterns 2, 5 and 6 have a consistency lower than 70%. Overall, the average consistency value across the 14 patterns reaches 88%.

Pattern.3 (Neutral)		**Pattern.12 (Positive)**		**Pattern.5 (Mixed)**		**Pattern.11 (Positive)**	
Entity	True Sentiment	Entity	True Sentiment	Entity	True Sentiment	Entity	True Sentiment
Brazil	Neutral	Kardashian	Positive	Cancer	Negative	Amy Adams	Positive
Facebook	Neutral	Katy Perry	Positive	Fever	Negative	Dallas	Positive
Oprah	Neutral	Beatles	Positive	Headache	Negative	Riyadh	Positive
Sydney	Neutral	Usher	Positive	McDonald	Neutral	Sam	Positive
Seattle	Neutral	Pandora	Positive	Xbox	Neutral	Miley Cyrus	Positive

Table 27: Example of three strongly consistent SS-Patterns (Patterns 3, 11, and 12) and one inconsistent SS-Pattern (Pattern 5), extracted from the STS-Gold Entity dataset.

5.6.1 Sentiment Consistency vs. Sentiment Dispersion

From the above, we observed that patterns 2, 5 and 6 have low sentiment consistency. Looking at the characteristics of entities in these patterns, we notice that the average dispersion of their SentiCircles is 0.18 on average. This is twice higher than the dispersion of the entities within the other 11 strongly consistent patterns. Overall, we found

a negative correlation of -0.42 between the sentiment consistency of SS-Patterns and the dispersion of their entities' SentiCircles. This indicates that SS-Patterns that contain entities of highly dispersed SentiCircles are more likely to have low sentiment consistency. Based on the SentiCircle model (Section 5.3), these high dispersed entities either occur very infrequently or occur in different contexts of different sentiment in the tweet corpus.

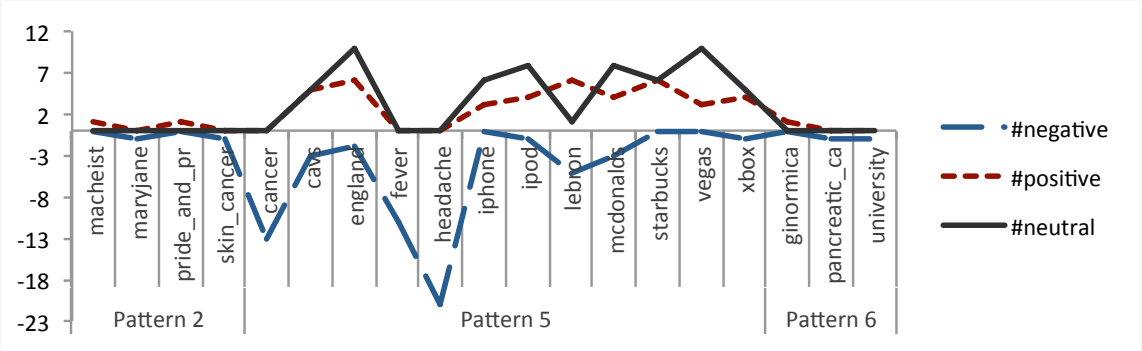


Figure 27: Number of times that entities in Patterns 2, 5 and 6 receive negative, positive and neutral sentiment in the STS-Gold dataset.

To validate our above observation, we analyse the human sentiment votes on the 58 entities in STS-Gold dataset.[5] Figure 27 shows entities under patterns 2, 5 and 6 along with the number of times they receive negative, positive and neutral sentiment in tweets according to the three human annotators. We observe that entities in patterns 2 and 6 occur very infrequently in tweets, yet with consistent sentiment. On the other hand, most entities in Pattern 5 occur more frequently in tweets. However, they receive controversial sentiment (i.e., opposite sentiment). For example, the entity "`McDonald's`" occurs 3, 4 and 8 times with negative, positive and neutral sentiment respectively.

The above analysis shows the potential of our approach for generating patterns of entities that indicate sentiment disagreement or controversy in tweets.

5.7 DISCUSSION

In this chapter we showed the value of our proposed approach in extracting semantic sentiment patterns of words and using them for sentiment analysis on Twitter. We aimed to address the question of whether incorporating SS-Patterns as features into supervised

5 Human votes on each entity are available to download with the STS-Gold dataset under http://tweenator.com

classifier training improves the sentiment classification performance at tweet and entity levels.

Our patterns, by definition, are based on words' similarities in a given context in tweets, which makes them relevant to that specific context. This means that they might need updating more frequently than context-independent patterns, i.e., patterns derived based on pre-defined syntactic templates [130] or common-sense knowledge bases [25]. Hence, a potential gain in performance may be obtained by combining context-independent patterns with our patterns, which introduces a future direction to this work.

For tweet-level sentiment classification, SS-Patterns were evaluated on 9 Twitter datasets with different results. For example, our SS-Patterns produced the highest performance improvement on the STS-Test dataset (+9.78% over the baseline) while the lowest improvement was obtained on the GASP dataset (+1.23%). Different factors might be behind such variance. For example, our datasets differ in their sizes, sparsity degrees and sentiment class distributions. A future direction of this work is to study the influence of these factors on (i) the quality of the extracted patterns and (ii) their usefulness in sentiment analysis. Analysing such influence potentially allows for adapting our extraction approach to the characteristics of the Twitter dataset analysed, enabling better and more accurate pattern extraction.

For entity-level sentiment classification, the evaluation was performed on one dataset and by using a single classifier. We noticed that detecting positive entities was easier than detecting neutral or negative entities. This might be due to: (i) the choice of the classifier we used, or (ii) the imbalanced sentiment class distribution in the dataset. As mentioned earlier, the number of positive entities is twice larger than the number of negative or neutral ones in the evaluation dataset. Also, our classifier is based on using the back-off strategy, which makes the incorporation of our patterns in sentiment classification limited only to those cases where entities do not occur in tweets of a certain sentiment class. Additional experimental work may help assess the effect of using balanced entity datasets and different types of entity sentiment classifiers on the performance of our patterns.

We showed that our approach was able to discover patterns of entities that could indicate sentiment disagreement or controversy in tweets. Those patterns have shown low consistency with the sentiment of entities within them. Thus, one may expect terms under these patterns to negatively impact sentiment classifiers, and therefore, removing them

from the feature space for sentiment classification might help improve the classification performance. This creates a room for more research in this direction.

5.8 SUMMARY

Sentiment is often expressed via latent semantic relations, patterns and dependencies among words in tweets. In this chapter we aimed to address the third research question of this thesis: *Could semantic sentiment patterns boost sentiment analysis performance?* To this end, we propose a novel approach that automatically captures patterns of words of similar contextual semantics and sentiment in tweets. Unlike previous work on sentiment pattern extraction, our proposed approach does not rely on external and fixed sets of syntactical templates/patterns, nor requires deep analyses of the syntactic structure of sentences in tweets.

We assessed the performance of our approach by first extracting SS-Patterns from several Twitter datasets. After that, we validated the extracted patterns by using them as classification features in both, tweet- and entity-level sentiment analysis tasks.

For tweet-level sentiment analysis we trained MaxEnt and NB classifiers from SS-Patterns using 9 different Twitter datasets. For entity-level sentiment analysis we designed our own sentiment classifier based on maximum likelihood estimation. The entity sentiment classifier was evaluated on a single dataset of 58 manually annotated named-entities.

Evaluation results showed that classifiers trained from our SS-Pattern in both sentiment analysis tasks and on all datasets produce a consistent and superior performance over classifiers trained from 6 state-of-the-art sets of features derived from both, the syntactic and semantic representations of words.

We conducted an analysis of our SS-Patterns and showed that the efficiency of our patterns in sentiment analysis is due to their strong consistency with the sentiment of terms within them. Also, the analysis showed that our approach was able to derive patterns of entities of controversial sentiment in tweets.

Part III

ANALYSIS STUDY

"We must think things not words, or at least we must constantly translate our words into the facts for which they stand, if we are to keep to the real and the true."

Oliver Wendell Holmes Jr

6

STOPWORD REMOVAL FOR TWITTER SENTIMENT ANALYSIS

In previous chapters, we chose not to remove stopwords in our preprocessing of Twitter data since stopwords tend to carry sentiment information as suggested by several works on Twitter sentiment analysis.

However, the impact of removing stopwords on the performance of Twitter sentiment classifiers remains debatable. Some works argue that stopword removal reduces the textual noise in tweets and consequently improves the sentiment classification performance.

In this chapter we study the impact of removing stopwords in the sentiment analysis task on Twitter. In particular, we investigate whether removing stopwords helps or hampers the effectiveness of Twitter sentiment classification methods. To this end, we apply six different stopword identification methods to Twitter data from six different datasets and observe how removing stopwords affects two well-known supervised sentiment classification methods. We assess the impact of removing stopwords by observing fluctuations on the level of data sparsity, the size of the classifier's feature space and its classification performance.

Our conclusion is that stopwords, in most cases, do carry sentiment information and removing them from tweets has a negative impact on the sentiment classification performance.

6.1 INTRODUCTION

The discussion brought in Chapters 1 and 2 showed that one of the key challenges that Twitter sentiment analysis methods have to confront is the noisy nature of Twitter generated data. Twitter allows only for 140 characters in each post, which influences the use of abbreviations, irregular expressions and infrequent words. This phenomena increases the level of data sparsity, affecting the performance of Twitter sentiment classifiers [137].

A well known method to reduce the noise of textual data is the removal of stopwords. This method is based on the idea that discarding non-discriminative words reduces the

feature space of the classifiers and helps them to produce more accurate results [145]. This pre-processing method, widely used in the literature of document classification and retrieval, has been applied to Twitter in the context of sentiment analysis obtaining contradictory results. While some works support their removal [10, 110, 185, 152, 61, 82, 5] others claim that stopwords indeed carry sentiment information and removing them harms the performance of Twitter sentiment classifiers [136, 73, 96, 72].

In addition, most of the works that have applied stopword removal for Twitter sentiment classification use general stopwords lists, such as the *Van stoplist* [129], the *Brown stoplist* [54], etc. However, these stoplists have been criticised for: (i) being outdated [92, 149] (a phenomena that may affect specially Twitter data, where new information and terms are continuously emerging) and, (ii) for not accounting for the specificities of the domain under analysis [7, 182], since non-discriminative words in one domain or corpus may have discriminative power in different domain.

Aiming to solve the aforementioned limitations, several approaches have emerged in the areas of document retrieval and classification that aim to dynamically build stopword lists from the corpus under analysis. These approaches measure the discriminative power of terms by using different methods including: the analysis of terms' frequencies [163, 92], the term entropy measure [149, 148], the Kullback-Leibler (KL) divergence measure [92], and the Maximum Likelihood Estimation [7]. While these techniques have been widely applied in the areas of text classification and retrieval, their impact in Twitter sentiment analysis has not been deeply investigated.

In this chapter we aim to fill the above gap, by investigating the impact of stopword removal in the sentiment analysis task on Twitter and whether removing stopwords affects the performance of Twitter sentiment classifiers.

The research question we aim to address in this Chapter is:

RQ4 ***What effect does stopword removal have on sentiment classification performance?***

To address the above question, we apply six stopword removal methods over Twitter data from six different datasets (obtained from the literature of Twitter sentiment classification) and observe how removing stopwords affects polarity classification of tweets (positive vs. negative). To this end, we use two well-known supervised sentiment classification methods, Maximum Entropy (MaxEnt) and Naive Bayes (NB). We assess the impact of removing stopwords by observing fluctuations on: (i) the level of data sparsity, (ii) the size of the classifier's feature space and (iii), the classifier's performance in terms of accuracy and F-measure.

Our results show that general stopword lists (classic stoplists) indeed hamper the performance of Twitter sentiment classifiers. Regarding the use of dynamic methods, stoplists generated by mutual information produce the highest increase in the classifier's performance compared to not removing stopwords (1.78% and 2.54% average increase in accuracy and F-measure respectively) but a moderate reduction on the feature space and with no impact on the data sparsity. On the other hand, removing singleton words (those words appearing only once in the corpus) maintain a high classification performance while shrinking the feature space by 65% and reducing the dataset sparsity by 0.37% on average. Our results also show that while the different stopword removal methods affect sentiment classifiers similarly, Naive Bayes classifiers are more sensitive to stopword removal than the Maximum Entropy ones.

The rest of the Chapter is organized as follows. Our analysis set up is presented in Section 6.2. Results obtained from our analysis are presented in Section 6.3. Discussion is covered in Section 6.4. Finally, we summarise our work in in this chapter in Section 6.5.

6.2 STOPWORD ANALYSIS SET-UP

As mentioned in the previous section, our aim is to assess how different stopword removal methods affect the performance of Twitter sentiment classifiers. To this end, we assess the influence of six stopword removal methods over six Twitter corpora, using two sentiment classifiers. These components are detailed in the following subsections.

6.2.1 Datasets

Stopwords may have different impact in different contexts. Words that do not provide any discriminative power in one context may carry some semantic information in another context. In this chapter we study the effect of stopword removal in six different Twitter datasets obtained from the literature of Twitter sentiment classification:

- The Obama-McCain Debate dataset (OMD) [144].
- The Health Care Reform data (HCR) [152].
- The STS-Gold dataset [138].
- Two datasets from the Dialogue Earth project (GAS, WAB) [5].

- The SemEval dataset [104].

Table 28 shows the total number of tweets and the vocabulary size (i.e., number of unique word unigrams) within each dataset. Note that we only consider the subsets of positive and negative tweets from these datasets since we perform binary sentiment classification (positive vs. negative) in our analysis.[1]

Dataset	No. of Tweets	Vocabulary Size
OMD	1,081	3,040
HCR	1,354	5,631
STS-Gold	2,034	5,780
SemEval	5,387	16,501
WAB	5,495	10,920
GASP	6,285	12,828

Table 28: Statistics of the six datasets used in evaluation.

6.2.2 Stopword removal methods

The *Baseline method* for this analysis is the non removal of stopwords. In the following subsections we introduce the stopword removal methods we use in our study.

6.2.2.1 *The Classic Method*

This method is based on removing stopwords obtained from general lists. Multiple lists exist in the literature [129, 54]. But for the purpose of this work we have selected the classic Van stoplist [129].

6.2.2.2 *Methods based on Zipf's Law (Z-Methods)*

In addition to the classic stoplist, we use three stopword generation methods inspired by Zipf's law[2] [188] including: removing most frequent words (*TF-High*) [92, 163] and re-

1 Details about the construction and the annotations of these datasets are provided in Appendix A

2 Zipf's law [188] states that in a data collection the frequency of a given word is inversely proportional to its rank.

moving words that occur once, i.e. singleton words (*TF1*) [93]. We also consider removing words with low inverse document frequency (*IDF*) [53, 92].

To choose the number of words in the stoplists generated by the aforementioned methods, we first rank the terms in each dataset based on their frequencies (or the inverse document frequencies in the IDF method). Secondly, we plot the rank-frequency distribution of the ranked terms. The size of the stoplist corresponds to where an "elbow" appears in the plot. For example, Figure 28 shows the rank-frequency distribution of terms in the GASP dataset with the upper and lower cut-offs of the elbow in the distribution plot. From this Figure, the TF-High stoplist is supposed to contain all the terms above the upper cut-off (50 terms approximately). On the other hand, the TF1 stoplist should contain all the terms below the lower cut-off, which occur only once in tweets.

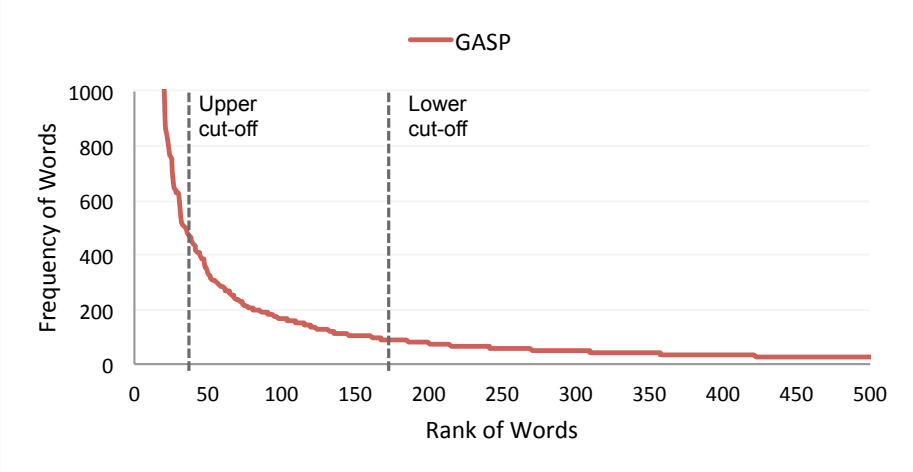


Figure 28: Rank-Frequency distribution of the top 500 terms in the GASP dataset. We removed all other terms from the plot to ease visualisation.

Figure 29 shows the rank-frequency distribution of terms for all datasets in a log-log scale. Although our datasets differ in the number of terms they contain, one can notice that the rank-frequency distribution in all the six datasets fits well the Zipf distribution.

6.2.2.3 *Term Based Random Sampling (TBRS)*

This method was first proposed by Lo et al. [92] to automatically detect stopwords from web documents. The method works by iterating over separate chunks of data randomly selected. It then ranks terms in each chunk based on their informativeness values using the Kullback-Leibler divergence measure [41] as shown in Equation 21.

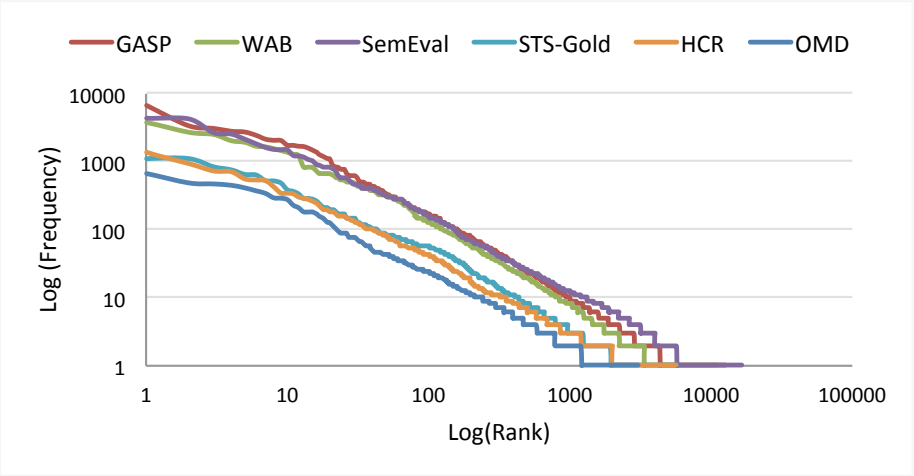


Figure 29: Frequency-Rank distribution of terms in all the datasets in a log-log scale.

$$d_x(t) = P_x(t).\log_2 \frac{P_x(t)}{P(t)} \tag{21}$$

where $P_x(t)$ is the normalised term frequency of a term t within a chunk x, and $P(t)$ is the normalised term frequency of t in the whole collection.

The final stoplist is then constructed by taking the least informative terms in all chunks, removing all possible duplications.

6.2.2.4 *The Mutual Information Method (MI)*

Stopwords removal can be thought of as a feature selection routine, where features that do not contribute toward making correct classification decisions are considered stopwords and got removed from the feature space. The mutual information method (MI) [41] is a supervised method that works by computing the mutual information between a given term and a document class (e.g., positive, negative), providing an indication of how much information the term can tell about a given class. Low mutual information suggests that the term has low discrimination power and hence it should be removed.

Formally, the mutual information between two random variables representing a term t and a class c is calculated as [180]:

$$I(T;C) = \sum_{t \in T} \sum_{c \in C} p(t,c) \log \left(\frac{p(t,c)}{p(t).p(c)} \right) \tag{22}$$

Where $I(T;C)$ denotes the mutual information between T and C and $T = \{0,1\}$ is the set in which a term t occurs ($T = \{1\}$) or does not occur ($T = \{0\}$) in a given document.

$C = \{0, 1\}$ represents the class set of the document. If the document belongs to class c then $C = \{1\}$, otherwise $C = \{0\}$.

Note that the size of the stoplists generated by both the MI and the TBRS methods is determined using the elbow approach as in the case of Z-Methods, i.e., ordering terms with respect to their informativeness values and search for where the elbow appears in the rank-informativeness plot.

6.2.3 Twitter Sentiment Classifiers

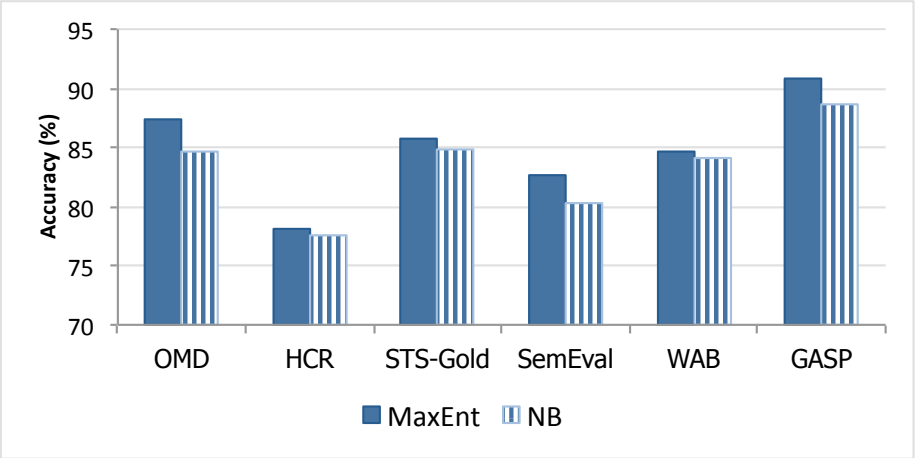
To assess the effect of stopwords in sentiment classification we use two of the most popular supervised classifiers used in the literature of sentiment analysis, Maximum Entropy (MaxEnt) and Naive Bayes (NB) from Mallet.[3] We report the performance of both classifiers in accuracy and average F-measure using a 10-fold cross validation. Also, note that we use unigram features to train both classifiers in our experiments.

6.3 EVALUATION RESULTS

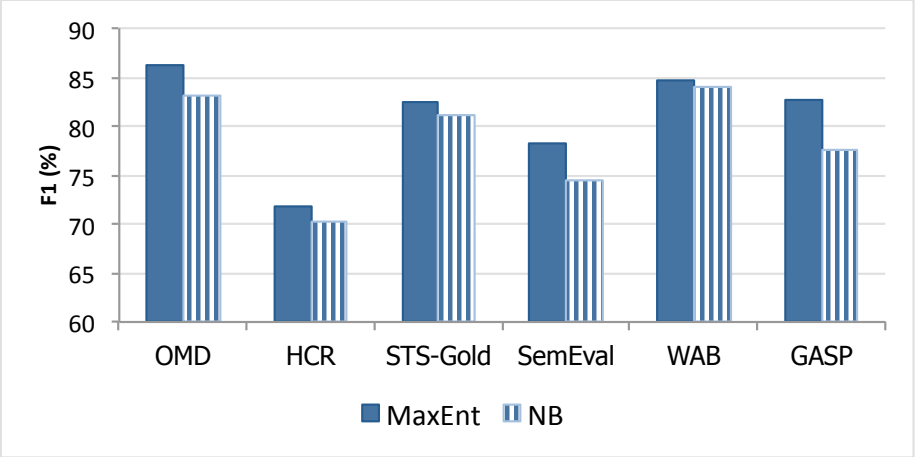
To study the effect of stopword removal in Twitter sentiment classification we apply the previously described stopword removal methods and assess how they affect sentiment polarity classification (positive / negative classification of tweets). We assess the impact of removing stopwords by observing fluctuations (increases and decreases) on three different aspects of the sentiment classification task: the *classification performance*, measured in terms of accuracy and F1-measure (F1), the size of the classifier's *feature space* and the level of *data sparsity*. Our baseline for comparison is not removing stopwords.

Figure 30 shows the baseline classification performance in accuracy (a) and F-measure (b) for the MaxEnt and NB classifiers across all the datasets. As we can see, when no stopwords are removed, the MaxEnt classifier always outperforms the NB classifier in accuracy and F1 measure on all datasets.

3 http://mallet.cs.umass.edu/



(a) Average Accuracy.



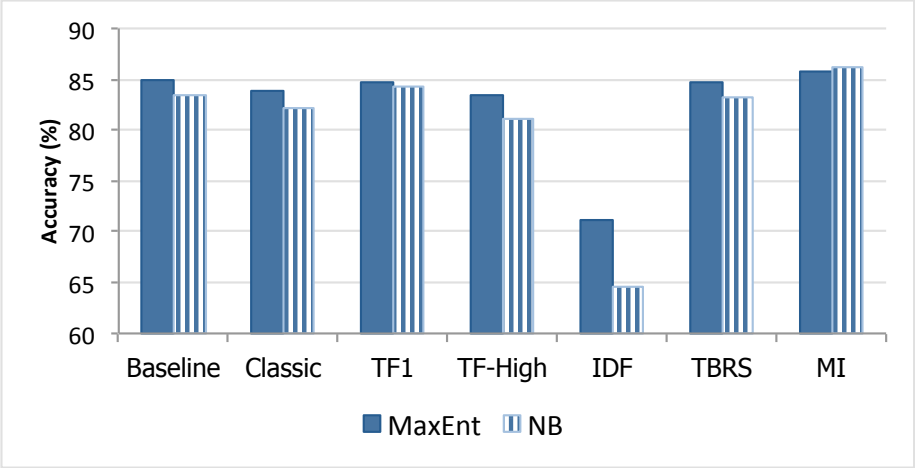
(b) Average F-measure.

Figure 30: The baseline classification performance in Accuracy and F1 of MaxEnt and NB classifiers across all datasets.

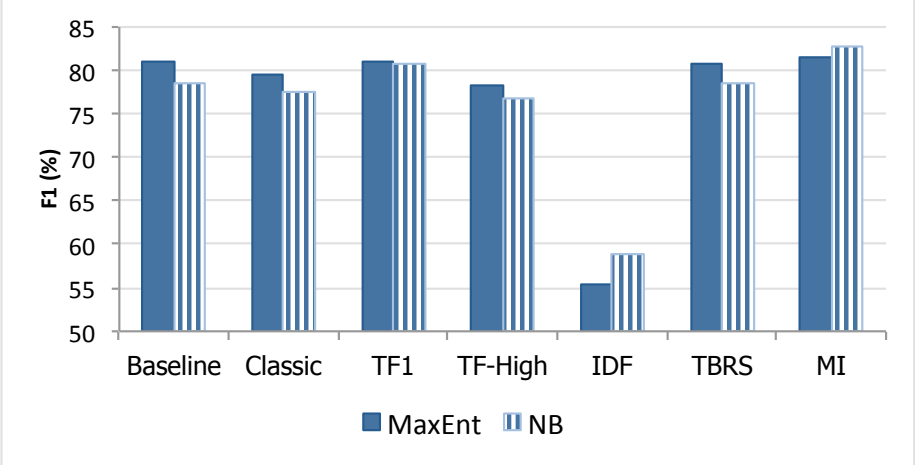
6.3.1 Classification Performance

The first aspect that we study is how removing stopwords affects the classification performance. Figure 31 shows the average performances across all datasets in accuracy (Figure 31:a) and F1 (Figure 31:b) obtained from the MaxEnt and NB classifiers by using the previously described stopword removal methods. A similar performance trend can be observed for both classifiers. For example, a significant loss in accuracy and in F1 is encountered when using the IDF stoplist, while the highest performance is always obtained when using the MI stoplist. It also worth noting that using the classic stoplist gives lower performance than the baseline with an average loss of 1.04% and 1.24% in accuracy and

F-measure respectively. Removing singleton words (the TF1 stoplist) improves accuracy by 1.15% and F1 by 2.65% compared to the classic stoplist. However, we notice that the TF1 stoplist gives 1.41% and 1.39% lower accuracy and F1 than the MI stoplist respectively. Nonetheless, generating TF1 stoplists is a lower cost process than generating MI stoplists, since, no training from labelled data is required.



(a) Average Accuracy.



(b) Average F-measure.

Figure 31: Average Accuracy and F-measure of MaxEnt and NB classifiers using different stoplists.

It can be also shown that removing the most frequent words (TF-Hight) hinders the average performance for both classifiers by 1.83% in accuracy and 2.12% in F-measure compared to the baseline. The TBRS stoplist seems to outperform the classic stoplist, but it just gives a similar performance to the baseline.

Finally, it seems that NB is more sensitive to removing stopwords than MaxEnt. NB faces more dramatic changes in accuracy than MaxEnt across the different stoplists. For example, compared with the baseline, the drop in accuracy in NB is 9.8% higher than in MaxEnt when using the IDF stoplist.

6.3.2 Feature Space

The second aspect we study is the average reduction rate on the classifier's feature space caused by each of the studied stopword removal methods. Note that the size of the classifier's feature space is equivalent to the vocabulary size for the purpose of this study. As shown in Figure 32, removing singleton words reduces the feature space substantially by 65.24%. MI comes next with a reduction rate of 19.34%. On the other hand, removing the most frequent words (TF-High) has no actual effect on the feature space (a marginal 0.82%). All other stoplists reduce the number of features by less than 12%.

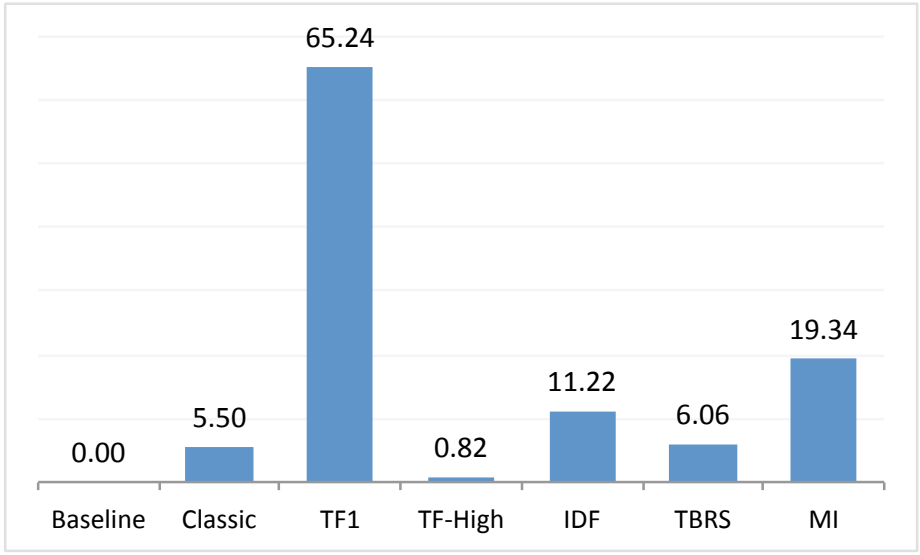


Figure 32: Reduction rate on the feature space of the various stoplists.

Two-To-One Ratio

As we have observed, removing singleton words reduces the feature space up to 65% on average. To understand what causes such a high reduction we analysed the number of singleton words in each dataset individually. As we can see in Figure 33 singleton words constitute two-thirds of the vocabulary size of all datasets. In other words, the ratio of singleton words to non singleton words is two to one for all datasets. This two-to-

one ratio explains the large reduction rate in the feature space when removing singleton words.

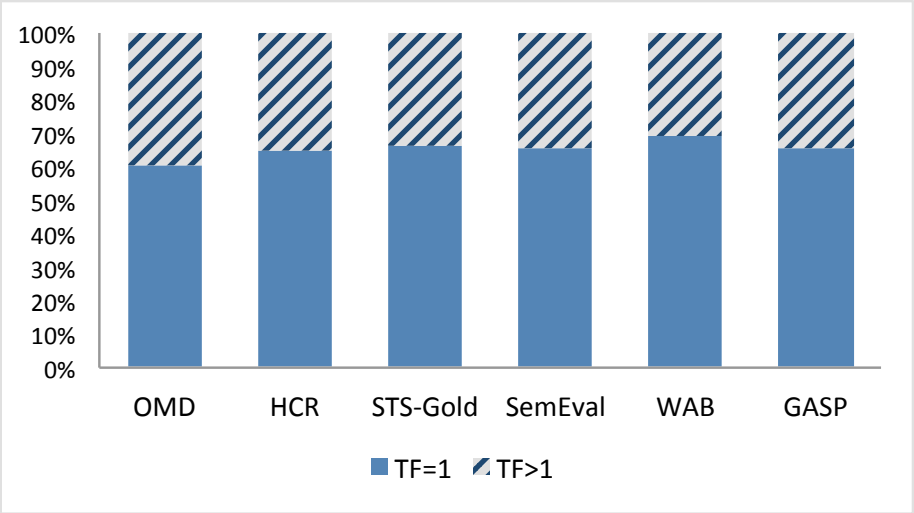


Figure 33: The number of singleton words to the number non singleton words in each dataset.

6.3.3 Data Sparsity

Dataset sparsity is an important factor that affects the overall performance of a typical machine learning classifier [119]. In our previous work [137], we showed that Twitter data are sparser than other types of data (e.g., movie review data) due to the large number of infrequent words present within tweets. Therefore, an important effect of a stoplist for Twitter sentiment analysis is to help reduce the sparsity degree of the data.

To calculate the sparsity degree of a given dataset, we first construct the term-tweet matrix $G \in R^{m \times n}$, where m and n are the number of the unique terms (i.e., vocabulary size) and tweets in the dataset respectively. The value of an element $e_{i,j} \in G$ can be either 0 (i.e., the term i does not occur in tweet j) or 1 (i.e., the term i occurs in tweet j). According to the sparse nature of tweets data, matrix G will be mostly populated by *zero* elements.

The sparsity degree of G corresponds to the ratio between the number of the *zero* elements and total number of all elements [93] as follows:

$$S_d = \frac{\sum_j^n N_j}{n \times m} \tag{23}$$

Where N_j is the number of *zero* elements in column j (i.e., tweet j). Here $S_d \in [0, 1]$, where high S_d values refer to a high sparsity degree.

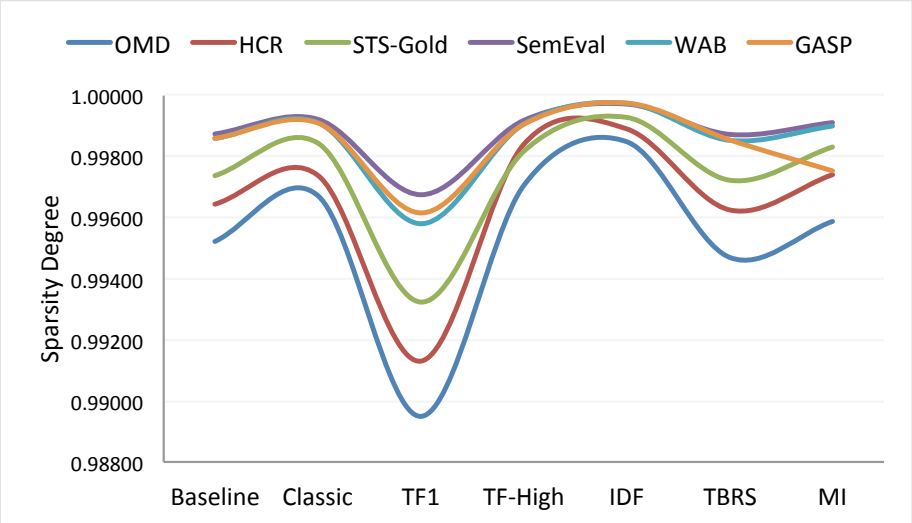


Figure 34: Stoplist impact on the sparsity degree of all datasets.

Figure 34 illustrates the impact of the various stopword removal methods on the sparsity degree across the six datasets. We notice that our Twitter datasets are very sparse indeed, where the average sparsity degree of the baseline is 0.997.

Compared to the baseline, using the TF1 method lowers the sparsity degree on all datasets by 0.37% on average. On the other hand, the effect of the TBRS stoplists is barely noticeable (less than 0.01% of reduction). It is also worth highlighting that all other stopword removal methods increase the sparsity effect with different degrees, including the classic, TF-High, IDF and MI.

From the above we notice that the reduction on the data sparsity caused by the TF1 method is moderate, although the reduction rate on the feature space is 65.24%, as shown in the previous section. This is because removing singleton words reduces the number of *zero* elements in G as well as the total number of elements (i.e., $m \times n$) at very similar rates of 66.47% and 66.38% respectively, as shown in Table 29. Therefore, the ratio in Equation 23 improves marginally, producing a small decrement in the sparsity degree.

6.3.4 The Ideal Stoplist

The ideal stopword removal method is the one which helps maintaining a high classification performance, leads to shrinking the classifier's feature space and effectively reducing the data sparseness. Moreover, since Twitter operates in a streaming fashion (i.e., millions of tweets are generated, sent and discarded instantly), the ideal stoplist method is required to have low runtime and storage complexity and to cope with the continuous

Method	Reduction(Zero-Elm)	Reduction(All-Elm)
Calssic	3.398	3.452
TF1	66.469	66.382
TF-High	0.280	0.334
IDF	0.116	0.117
TBRS	3.427	3.424
MI	27.810	27.825

Table 29: Average reduction rate on *zero* elements (Zero-Elm) and all elements (All-Elm) of the six stoplist methods.

shift in the sentiment class distribution in tweets. Lastly and most importantly, the human supervision factor (e.g., threshold setup, data annotation, manual validation, etc.) in the method's workflow should be minimal.

Table 30 sums up the evaluation results reported in the three previous subsections. In particular, it lists the average performances of the evaluated stoplist methods in terms of: (i) the sentiment classification accuracy and F1-measure, (ii) the reduction on the feature space and the data sparseness, (iii) and the type of the human supervision required. According to these results, the MI and the TF1 methods show very competitive performances comparing to other methods; the MI method comes first in accuracy and F1 while the TF1 method outperforms all other methods in terms of feature space and data sparseness reduction.

Recalling Twitter's special streaming nature and looking at the human supervision factor, the TF1 method seems a simpler and more effective choice than the MI method. First, because the notion behind TF1 is rather simple - "*stopwords are those which occur once in tweets*", and hence, the computational complexity of generating TF1 stoplists is generally low. Secondly, the TF1 method is fully unsupervised while the MI method needs human supervision, including: (i) deciding on the size of the generated stoplists, which is usually done empirically (see Section 6.2.2.4), and (ii) manually annotating tweet messages with their sentiment class label in order to calculate the informativeness values of terms as described in Equation 22.

Hence, in practical sentiment analysis applications on Twitter, where a large number of tweets is required to be classified at high speed, a marginal loss in classification per-

Stoplist	Accuracy	F1	R-Rate	Data Sparsity	Human Supervision
Classic	83.09	78.46	5.50	0.08	Unsupervised
TF1	84.50	80.85	**65.24**	**-0.37**	Unsupervised
TF-High	82.31	77.58	0.82	0.1	Threshold Setup
IDF	67.85	57.07	11.22	0.182	Threshold Setup
TBRS	84.02	79.60	6.06	-0.017	Threshold Setup
MI	**85.91**	**82.23**	19.34	0.037	Threshold Setup & Data Annotation

Table 30: Average accuracy, F1, reduction rate on feature space (R-Rate) and data sparsity of the six stoplist methods. Positive sparsity values refer to an increase in the sparsity degree while negative values refer to a decrease in the sparsity degree. Human Supervision refers to the type of supervision required by the stoplist method.

formance could be traded with simplicity, efficiency and low computational complexity. Therefore, the TF1 method is recommended for this type of application. On the other hand, in applications where the data corpus consists of a small number of tweets in a specific domain or topic, any gain in the classification performance is maybe favoured over the other factors.

6.4 DISCUSSION

We evaluated the effectiveness of using several of-the-shelf stopword removal methods for polarity classification of tweets data. Our evaluation involved 6 Twitter datasets and investigated the impact the stopword removal methods have on the level of data sparsity of these datasets, the size of the classifier's feature space, and the sentiment classification performance.

One factor that may affect our results is the choice of the sentiment classifier and the features used for classifier training. In this study we used MaxEnt and NB classifiers trained from word unigrams. Therefore, the use other machine learning classifiers (e.g., support vector machines and regression classifiers) as well as new sets of features (e.g., word n-grams, part-of-speech tags, microblogging features) creates room for further investigation.

Our study revealed that the ratio of singleton words to non singleton words is two to one for all the six datasets. As described earlier, this ratio explains the high reduction rate on the feature space when removing singleton words. However, further analysis on whether the two-to-one ratio can be generalised to any Twitter dataset is yet to be conducted. Such analysis can be done by randomly sampling a large amount of tweets over different periods of time and studying the consistency of our ratio along all data samples.

We showed the value of the TF1 method over other stopword removal methods. Nevertheless, this method as well as the classic method or other frequency-based methods, ignores the context of words when detecting their informative value in tweets. However, the informative value of words, in many cases, relies on their semantics and context, where words with low informative values in some context or corpus, may have discrimination power in a different context. For example, the word "`like`", generally considered as stopword, has an important sentiment discrimination power in the sentence "`I like you`". Thus, one may expect that methods, which consider words' semantics when generating stopword lists, are likely to outperform those who rely only on frequency measures. Thus, investigating the impact of semantics on stopword extraction and removal for sentiment analysis on Twitter introduces a direction for future research.

6.5 SUMMARY

In this chapter we investigated the issue of removing stopwords in Twitter sentiment analysis, aiming at addressing the fourth question of this thesis: *What effect does stopword removal have on sentiment classification performance?*. To this end, we studied how six different stopword removal methods affect the sentiment polarity classification on Twitter.

Our results suggested that, despite its popular use in Twitter sentiment analysis, the use of general (classic) stoplist has a negative impact on the classification performance. We also observed that, although the MI stopword generation method obtains the best classification performance, it has a low impact on both the size of the feature space and the dataset sparsity degree.

A relevant conclusion of this study is that the TF1 stopword removal method is the one that obtains the best trade-off, reducing the feature space by nearly 65%, decreasing the data sparsity degree up to 0.37%, and maintaining a high classification performance. In practical applications for Twitter sentiment analysis, removing singleton words is the

simplest, yet most effective practice, which keeps a good trade-off between good performance and low processing time.

Finally, results showed that while the different stopword removal methods affect sentiment classifiers similarly, Naive Bayes classifiers are more sensitive to stopword removal than the Maximum Entropy ones.

Part IV

CONCLUSION

It is difficult to be rigorous about whether a machine really 'knows', 'thinks', etc., because we are hard put to define these things. We understand human mental processes only slightly better than a fish understands swimming.

John McCarthy

7

DISCUSSION AND FUTURE WORK

In this thesis we investigated the extraction and use of words' semantics to enhance sentiment analysis performance on Twitter. The research methodology that we followed in our research investigation, as discussed in Chapter 1, consisted of the following three steps:

1. Semantic Extraction: In this step, we extracted the conceptual and contextual semantics of words from tweets. While we used existing tools to extract words' conceptual semantics (Section 4.2.1), we proposed a new semantic representation model for capturing words' contextual semantics (Section 3.2).

2. Semantic Incorporation: here, we proposed methods and models for incorporating and using both, the contextual and conceptual semantics of words in sentiment analysis (Section 3.3 and Section 4.2.2). We also proposed an approach for generating semantic patterns of words and using them to train sentiment classifiers (Section 5.3).

3. Assessment: in this step we tested the performance of our proposed methods in several sentiment analysis tasks on Twitter, we used multiple evaluation datasets, and we compared our models against several state-of-the-art sentiment analysis baselines (Sections 3.3.4, 4.2.4 and 5.5).

Results in most of the evaluation scenarios we experimented with demonstrated the effectiveness of using our proposed semantic approaches for sentiment analysis, in comparison with other non-semantic or traditional approaches. Nevertheless, semantic extraction and incorporation was not as trivial; we witnessed several issues and challenges to our proposed approaches that may have had some impact on their performance.

In this chapter we discuss the main challenges that our approaches faced under each of the above steps and present future work directions to tackle them.

7.1 DISCUSSION

7.1.1 Extracting Words' Semantics

The first task of our work in this thesis was extracting and capturing the semantics of words from Twitter data. To this end, we followed two research directions. The first one assumes that the semantics of words can be derived from words' co-occurrence patterns in tweets (aka, contextual semantics). The other direction suggests that the meaning of words is directly associated with their explicit semantic concepts (aka, conceptual semantics), which can be obtained from external knowledge sources, such as ontologies and semantic networks.

We proposed several approaches for extracting both type of semantics from tweets. Each comes with several opportunities for improvements, as will be discussed subsequently.

7.1.1.1 *Extracting Contextual Semantics*

Motivated by the distributional hypothesis; that words that occur in the same context tend to have similar meanings, we proposed an approach called SentiCircles that captures the contextual (or distributional) semantics of a word from its co-occurrence patterns in a given context (Chapter 3).

One thing that may affect the performance of our approach is the granularity or the level of the context, in which the semantics are extracted. In our case, the SentiCircle approach defines the context as a Twitter corpus or a collection of tweets. This means that our approach aims to extract the semantics of a word at the corpus level (e.g., `Trojan Horse` refers to `Computer Virus` in one corpus and refers to `Greek mythology` in another corpus), but not at the tweet or sentence level. However, there are cases where the semantics of a word changes from one tweet to another even in the same Twitter corpus. These cases are more likely to happen when the corpus under analysis is too generic and reflects multiple topics. Extracting fine-grained contextual semantics of a word in such corpora requires capturing the context by SentiCircles at the tweet and even the sentence level. One solution for this is to apply the SentiCircle approach to each tweet, where the word occurs, separately. This means building a SentiCircle for each occurrence of the word in each tweet. This solution may allow (i) distinguishing between the different contexts the word has in different tweets, and (ii) calculating the semantics of the word

with respect to each of these contexts independently. However, building multiple SentiCircles for each word in the corpus may increase the runtime complexity of our approach. Moreover, some of these SentiCircles may not carry enough semantic information due the lack of context in some tweets or due to their sparseness. Including these SentiCircles in the analysis may affect the performance of our approach. A fix to this issue could be by enriching words' SentiCircles with features extracted from the syntactic, linguistic or semantic representation of the terms within them (e.g., POS tags, Twitter features, LDA topics, etc.). Exploring these solutions and their impact on our approach creates room for future work.

7.1.1.2 *Extracting Conceptual Semantics*

The second type of semantics we investigated in this thesis is conceptual semantics, i.e., the semantic concepts (e.g., `Virus/Disease`) of named entities (e.g, `Ebola`) found in tweets (Chapter 4).

To extract this type of semantics we used several off-the-shelf semantic entity extractors including Zemanta,[1] OpenCalais[2] and AlchemyAPI[3] (see Section 4.2.1).

Unlike SentiCircle, which functions at corpus level, we ran these extractors on each tweet in our Twitter corpora individually, in order to extract the semantic concepts in each given tweet.

However, one thing that may impact the effectiveness of using this approach is the abstraction level of the semantic concepts retrieved. In many cases, these concepts were too abstract (e.g. Person) which were equally used for mentions of ordinary people, as well as for famous musicians or politicians (see Section 4.2.3.2). For the tweet "`i wish i could go to france and meet president Obama haha`", AlchemyAPI provided the concept *Person* to represent "president Obama", whereas Zemanta identified him with the concept */government/politician* which is more specific. Increasing the specificity of such concepts, perhaps with the aid of DBpedia, or using multiple entity extractors, might lead to better semantic incorporation and consequently to increasing the sentiment analysis performance.

1 http://www.zemanta.com/
2 http://www.opencalais.com/
3 http://www.alchemyapi.com/

7.1.1.3 *Extracting Stopwords*

Another aspect of improving the performance of sentiment classifiers on Twitter, is to find words that have a low informativeness value, which are also known as stopwords, and remove them from the analysis in order to reduce the noise of tweets and consequently improve sentiment analysis performance.

In Chapter 6, we investigated the problem of extracting and removing stopwords from tweets and its impact on sentiment analysis performance. Results showed that stopword removal, based on general stoplists, tends to hinder the performance of Twitter sentiment classifiers (see Section 6.3.1). Therefore, in this thesis we chose not to remove stopwords from our Twitter datasets.

However, our analysis in Chapter 6 showed that sentiment analysis may benefit from removing stopwords under specific conditions. For example, removing words with low mutual information (Section 6.2.2.4) or with low KL divergence values (Section 6.2.2.3) from tweets improved the classification performance by up to 2.8% over using general stopword lists. Investigating the impact of these stopword removal methods on the performance of our semantic approaches introduces a direction for future research.

7.1.2 Incorporating Words' Semantics in Sentiment Analysis

In the second task of our work, we proposed several methods for using both, the contextual and conceptual semantics of words with lexicon-based and supervised machine learning sentiment analysis approaches in order to enhance their performance (check Sections 3.3, 4.2.2 and 5.4), as will be discussed subsequently.

7.1.2.1 *Incorporating Semantics into Lexicon-based Approaches*

1. Incorporating Contextual Semantics: We proposed several lexicon-based methods, based on SentiCircle, for entity- and tweet-level sentiment analysis, as described in Section 3.3.

Since SentiCircle functions at the corpus level, the sentiment assigned by the Median method (see Section 3.3.1) to an entity represents its aggregated sentiment score in the whole corpus. Hence, our method is not tuned to identifying the sentiment of an entity within a specific tweet in the corpus analysed. As discussed earlier in Section 7.1.1.1, a resolution for this issue is to consider applying the SentiCircle approach to the specific

tweet where the entity occurs. This in turn may allow our proposed methods to capture the sentiment of entities at the tweet level rather than at the corpus level.

To calculate the overall sentiment of a tweet, our approach updates the sentiment orientation and strength of all words in the tweet (Section 3.3.2). However, it might be the case, where the sentiment of some terms in different tweets is often stable and does not need to be changed by changing context. For example, both occurrences of the word "`excellent`" in the tweet "`excellent overview: The State of #JavaScript in 2015..`", and the tweet "`good friends. excellent food - High End fashion show...`" have positive sentiment regardless of the different contexts of this word in both tweets. A future work could be studying which type of terms change their sentiment, and which ones are more stable. This might help enhance the runtime performance of our approach by filtering out stable terms from the analysis. One way in which we may be able to identify a stable term is, for example, by looking at its sense(s) in WordNet since the term's associated sentiment might be related to its sense in a certain text. For example, the word "`excellent`" has one sense only in WordNet, and as such, its associated sentiment should be stable. On the other hand, the word "`great`" has 7 senses (e.g., "`large`", "`outstanding`", "`heavy`"), which often results in different sentiments due to these different senses.

2. *Incorporating Conceptual Semantics:* To incorporate conceptual semantics into lexicon-based sentiment analysis we investigated adding conceptual semantics to the SentiCircle representation and studying their impact on the overall sentiment detection performance (Section 4.3.1). In general, the detection performance in F-measure increased over using the raw SentiCircle representation. However, a marginal loss in accuracy was observed in two out the three datasets we experimented with. This might be due to the generality of some of the extracted concepts (e.g.,"Person", "Company"), which were applied to many terms of opposite sentiment. These concepts were regarded as normal terms in tweets, and had their own SentiCircles, which might have had a negative impact on the extraction of sentiment. A future direction might be to extract more specific concepts, using other concept extraction methods, as described in Section 7.1.1.2.

7.1.2.2 *Incorporating Semantics into Machine Learning Approaches*

We incorporated both, the contextual and conceptual semantics of words by using them as features to train different supervised classifiers. Incorporating conceptual semantics was realised by means of the semantic concepts of named entities in tweets (see Chapter

4). On the other hand, we incorporated the contextual semantics of words by using the patterns (semantic patterns) they belong to as training features (see Chapter 5).

The choice of the sentiment classifier is one factor that affects our analysis. In this thesis, we chose to incorporate our features into two machine learning classifiers; Naive Bayes and Maximum Entropy. Our choice was based on the efficiency, usability and popularity of these classifiers in previous works on sentiment analysis. However, machine learning classifiers are not limited to these two only, as shown in Chapter 2. For example, sentiment classification, based on support vector machines (SVM), have shown to outperform both, NB and MaxEnt classifiers in some previous works (Section 2.2.1.1). Testing our features against these classifiers creates room for future research.

Another factor that impacts our analysis is the incorporation method used. In Section 4.2.2 we proposed incorporating conceptual semantics into NB classifier using three methods; by replacement, augmentation and interpolation. The interpolation method proved its efficiency over all other methods. However, in Chapter 5 we used the augmentation method instead of the interpolation method for incorporating contextual semantic patterns. This is because the interpolation method is designed to work with Bayesian models, and in our experiments in Chapter 5 we used MaxEnt classifier since it showed to outperform NB classifier, when trained from contextual semantic features.

In all incorporation methods, all extracted semantic features were added to the analysis. However, it is likely that the contribution of each feature towards predicting the correct sentiment of a tweet or a word differs from other features. As such, identifying and filtering those features of low contribution (by using the information gain criterion, for example) might help shrink the feature space of classifiers besides improving the sentiment classification performance.

7.1.2.3 *Words' Semantics for Adapting Sentiment Lexicons*

In Chapter 3 we investigated the use of contextual semantics to adapt general-purpose sentiment lexicons. To this end, we proposed a rule-based method, based on SentiCircle, to amend the prior sentiment of words in Thelwall-Lexicon with their contextual sentiment, as explained in Section 3.4.

The evaluation of our adaptation method was done by analysing the performance of the adapted lexicon against the original one in tweet-level polarity classification (positive vs. negative), as explained in Section 3.4.1. However, our proposed method can assign

neutral sentiment to words. Therefore, a future direction might be to evaluate the adapted lexicon in subjectivity classification (subjective vs. neutral).

We performed a quantitative analysis on the adapted lexicon and showed that 96% of the words in Thelwall-Lexicon were affected by our adaptation method, i.e., these words were assigned new sentiment orientations and/or new sentiment strength (see Section 3.4.1, Table 9). A direction for future research is to conduct a qualitative analysis in order to check the percentage of these words that were correctly assigned contextual sentiment to those that were not.

7.1.3 Assessment and Results

In the experiments conducted throughout this thesis, we tested our proposed approaches in three popular sentiment analysis tasks on Twitter, we used several Twitter datasets and sentiment lexicons, and compared against several state-of-the-art methods. Below we discuss the main issues we faced in our assessment.

Evaluation Datasets: Evaluation at the tweet-level was done on multiple datasets of varying sizes, and topical-focus. Overall, results showed that our semantic approaches improved sentiment analysis performance upon other baseline methods we experimented with. However, the results in some parts were not as conclusive (see Sections 3.3.4 and 4.2.4). This is probably due to the characteristics of the dataset used, combined with the type of the semantics extracted. For example, our evaluation in Chapter 3 suggested that extracting and using contextual semantics in sentiment analysis is more efficient and accurate when the dataset analysed is of a specific topic focus. This is not surprising since our extraction approach, as mentioned earlier, looks at the general context (topic), which the dataset represents, when extracting words' contextual semantics and sentiment. On the other hand, our conceptual semantic approach (Chapter 4) seems to perform better on datasets of larger sizes and general diverse topic coverage. These observations are yet to be empirically confirmed by, for example, applying our approaches to more datasets and examine the consistency of their performance patterns against the datasets' properties.

The sentiment class distribution of the dataset also seems to impact the performance of our approaches, especially when extracting the sentiment of individual words and named entities. Specifically, results in Chapters 3, 4 and 5 showed that sentiment classification performance tends to be better with instances (i.e., tweets and entities) of sentiment labels

belonging to the dominant sentiment class in the dataset analysed. This is not surprising given that the sentiment class distribution in all our datasets is imbalanced. However, we believe that the evaluation of our approaches on the datasets used in this thesis gives a good impression of the performance we would expect by using our approaches in real life scenarios. This is because in a microblogging platform like Twitter, sentiment analysis approaches are likely to face chunks of tweets of skewed sentiment class distribution rather than balanced ones. Moreover, baselines in all our experiments were also evaluated on the same imbalanced datasets used to evaluate our approaches. In most cases, our approaches were found to outperform these baselines, which also denotes the effectiveness of our approaches over other baselines on such datasets.

A room for future work is to evaluate the proposed approaches in this thesis with more balanced datasets in order to assess their performance.

Analysis of Neutral Sentiment: In Chapter 3, our analysis at entity-level focused on polarity (i.e., positive vs. negative) and subjectivity (polar vs. neutral) detections. The sentiment of an entity in our work is considered neutral if the entity occurs either with no polar words in tweets, or with approximately an equal number of positive and negative words. Although the convention of neutral sentiment is typically defined based on these two cases, few applications and research works in the literature use the latter case as an indication of words/entities that reflect sentiment disagreement or controversy in tweets. Therefore, room for future work is to assess the performance of our approach on each case separately.

Baseline Methods: Cross-comparison in our evaluation was mostly done against non-semantic sentiment analysis baselines that are widely used on Twitter. Evaluation against semantic baselines was limited to those that rely on contextual semantics (e.g., LDA baselines). We did not compare against existing conceptual semantic baseline methods (e.g., Sentilo [126], Sentic Computing [25]) since these methods are not tailored to Twitter data (see Chapter 2), and therefore, evaluation results would have been biased toward our semantic approaches.

7.2 FUTURE WORK

Based on the discussion presented in the preceding section, we summarise our plan for future work by introducing the following research questions:

- **Does using more fine-grained semantics enhance sentiment analysis performance further?**

  As mentioned before, our semantic extractors in this thesis work at a very generic level, i.e., contextual semantics were extracted at the corpus level, and conceptual semantics were too abstract (person, company, etc.). Hence, the first question to be addressed beyond the work of this thesis is whether extracting and incorporating more specific and fine-grained semantics in our sentiment analysis approaches leads to increases in their performance.

  To answer this question, one would first improve the performance of the semantic extractors used in this thesis. For example, a future version SentiCircle should allow for contextual semantic extraction at tweet- and sentence-level. As explained in Section 7.1.1.1, this can be done by applying the SentiCircle approach on individual tweets rather than on the whole corpus. Also, enriching SentiCircle with words' syntactic or semantic features (e.g., POS tags, word synonymous, topic features, etc.), considering words' order in tweets, and looking at social context (e.g., the demographic and location information about the tweet's poster, the number of the poster's followers/followee, etc.), may all help capturing the specific semantics and sentiment of words in a given tweet or sentence.

  Also, new tools for extracting specific semantic concepts should be used or even developed rather than using third-party commercial tools that provide extracting generic and abstract concepts only. Named-entity recognition (NER) on Twitter, based on topic modelling, such as [131], or based on machine learning classifiers, such as [168], can be used to this end. As for named-entity linking (i.e., mapping entities to their proper semantic concepts in a given knowledge base), recent state-of-the-art approaches, such as YODIE [43],[4] can be used for this task.

- **How to use semantics in more-fine grained sentiment analysis subtasks and levels?**

4 https://gate.ac.uk/applications/yodie.html

Sentiment analysis is not limited to polarity and subjectivity detections, nor functions at tweet- and entity-level only, but also breaks down into more fine-grained subtasks and levels, such as detecting emotional states in text (e.g, "anger", "joy", "excitement", etc.), and sentiment of aspects (i.e., detecting the sentiment of products' aspects) (see Section 2.1). However, most existing approaches to these problems, as discussed in Chapter 2, are still non-semantic.

A room for future work is to investigate new techniques for updating our semantic approaches to perform emotion and aspect-level sentiment analysis tasks. For emotion detection, external emotion lexicons, such as [100], can be used to detect the type and intensity of emotions expressed via words in tweets. This information can be used in our approaches in different ways. For example, the prior sentiment of words in our SentiCircle representation (Section 3.2.2.2) can be replaced with the words' prior emotion score. Also, for our supervised sentiment classification approaches (Section 4.2.2), emotion information can be used as features together with semantic concepts to train sentiment classifiers.

As for aspect-level sentiment detection, detecting the sentiment of an aspect in a tweet requires applying several text processing, and semantic representation and reasoning techniques (e.g., representing the tweet using semantic networks or conceptual graphs) in order to define the syntactic and semantic roles of the words in that tweet (e.g., who is the opinion holder? what is the targeted object or aspect? what is the sentiment expressed towards that aspect?).

- **Do our semantic sentiment approaches generalise to other Mircoblogging platforms?**

  In this thesis, we experiment with Twitter data as a case study of microblogging data. Our approaches should be theoretically tailorable to data from other microblogging platforms and social networks (e.g., Facebook, Tumblr, etc), since they all share similar characteristics (e.g., the heavy use abbreviations, malformed words and emotions, lack of sentence structures, etc.). However, a proper experimental work is yet to be conducted in order to empirically check the applicability, and performance of our approaches when using data from these platforms in comparison to Twitter.

  A prerequisite for such investigation is the availability of gold-standard microblogging corpora required for evaluation. Currently, the number of such corpora in the

literature is quite limited. This is because the sentiment analysis problem on non-Twitter microblogging platforms (e.g., Facebook) is far less popular than the problem of sentiment analysis on Twitter. Building and annotating such gold-standards, although challenging, might be therefore required.

Posts generated on different microblogging platforms often have different character limits. For example, unlike the 140-character limit on Twitter, Facebook allows to share status updates up to 63,206 characters long.[5] Long status updates may contain multiple paragraphs, sentences or phrases of probably different sentiment orientations. Therefore, our approaches might need updating before being applied to such cases. One method to extract the overall sentiment of a long post in general is by dividing it into set of sentences, detecting the sentiment of each sentence solely, and averaging after that the sentiment of all these sentences.

- **Could our semantic approaches be used for stream-based sentiment analysis?**

  Our approaches were designed and tested to work in off-line mode, that is, by extracting sentiment from Twitter corpora of fixed sizes and limited number of tweets. In the real word, however, Twitter works in streaming fashion, where tweets are generated, posted and discarded in real time. This might be challenging to our approaches since they need to be constantly fed with new tweets to keep track of potential sentiment drifts in Twitter streams and provide up-to-date sentiment analysis. Hence, another potential future direction is to adapt our approaches to work in streaming mode. This requires certain alteration and optimization in order reduce their time-complexity and improve their scalability. For our SentiCircle approach, the size of the Term-Index used for building SentiCircles (see Section 3.2.2) should be kept small. This can be done by keeping only trending terms in the stream and removing fading ones. Also, words' SentiCircles need to be updated directly with new terms rather than re-computing them.

  For our machine learning models (Chapters 4 and 5), classifier learning should be done incrementally over data, meaning that, our classifiers must be constantly retrained with the continuous arrival of new data. In order to lower the reliance large labelled tweets for classifier retraining, we can start with a small training dataset and then continuously augment new labelled tweets in the training dataset. New

5 http://mashable.com/2011/11/30/facebook-status-63206-characters/

labelled tweets, in turn, can be obtained in different ways, such as using association rules [3, 146] or distant supervision [60] (see Chapter 2).

- **How to improve evaluation of our approaches?**

  Several issues, as discussed in the previous section, may impact our evaluation such as the choice of evaluation datasets, the type of sentiment analysis approach and evaluation baselines. Although a rigours evaluation was performed in this thesis, room for improving the assessment of our approaches would be by considering the following evaluation settings:

  - use new datasets of large size and balance sentiment class distribution.
  - design and experiment with our semantics using hybrid or ensemble classifiers that use multiple learning algorithms for sentiment analysis.
  - compare against more semantic sentiment analysis baselines.

8

CONCLUSION

The main goal of this thesis was to enhance sentiment analysis performance of microblogs. To achieve this goal we chose Twitter as a representative case study of microblogging platforms and investigated the role of words' semantics on the performance of multiple approaches to sentiment analysis on Twitter. To this end, we investigated the following four research questions:

RQ1 Could the *contextual semantics* of words enhance lexicon-based sentiment analysis performance?

RQ2 Could the *conceptual semantics* of words enhance sentiment analysis performance?

RQ3 Could *semantic sentiment patterns* boost sentiment analysis performance?

RQ4 What effect does stopword removal have on sentiment classification performance?

Our main intuition/motivation behind all our research work in this thesis is that traditional approaches to sentiment analysis on Twitter are semantically weak, since they do not account for the semantics of words when calculating their sentiment. This usually limits sentiment analysis performance, given that the sentiment of words is often associated with their semantics within the context they occur, as described in Chapter 2.

In the first part of this thesis (Chapters 3, 4, 5) we investigated different approaches for extracting and incorporating contextual and conceptual semantics into both machine learning and lexicon-based sentiment analysis approaches. We experimented with our approaches in three sentiment analysis tasks on Twitter; tweet-level sentiment analysis; entity-level sentiment analysis and context-aware sentiment lexicon adaptation. To this end, we used multiple Twitter datasets (check Appendix A) and sentiment lexicons, and compared the performance of our approaches against several state-of-the-art baselines. In the second part (Chapter 6) we turned our attention towards studying the effectiveness of several stopword removal methods for sentiment analysis of tweets and explored whether removing those words of weak semantics or low informativeness values (stopwords) in a given context influences the performance of Twitter sentiment classifiers.

Our main conclusion in this thesis is that the semantics of words should be considered when calculating their sentiment or the sentiment of tweets they belong to. Approaches that extract and use words' semantics for sentiment analysis surpass those that merely rely on affect words, or syntactic or linguistic structures that unambiguously reflect sentiment in tweets.

In this Chapter we summarise our main conclusions, contributions and findings with regards to our work under each of the above four research questions.

8.1 CONTEXTUAL SEMANTICS FOR SENTIMENT ANALYSIS OF TWITTER

In Chapter 3, we explored extracting and using the contextual semantics of words in lexicon-based sentiment analysis on Twitter. We aimed to answer the first research question of this thesis:

RQ1 ***Could the contextual semantics of words enhance lexicon-based sentiment analysis performance?***

To this end, we proposed SentiCircle, a semantic representation model that automatically captures the contextual semantics of a word from its co-occurrence patterns in a given Twitter corpus, and updates its sentiment orientation and sentiment strength respectively.

Based on SentiCircles, we built a lexicon-based approach to detect the sentiment at both, entity- and tweet-level. We evaluated our approach using 3 Twitter datasets and 3 sentiment lexicons, and compared its performance against three baseline methods in both tasks. Results showed that our approach at the entity-level outperformed all other baselines by 30-40% for subjectivity detection, and by 2-15% for polarity detection in average F-measure.

For tweet-level sentiment detection, results were competitive but inconclusive when comparing to state-of-art SentiStrength, and varied from one dataset to another. SentiCircle outperformed SentiStrength in accuracy by 4-5% in two datasets, and in F-measure by 1% in one dataset only.

We also proposed a rule-based approach, based on SentiCircles to amend the prior sentiment orientations and strengths of words in general-purpose sentiment lexicons with respect to the words' contextual sentiment. Evaluation was done on a single sentiment

lexicon using three Twitter datasets. Results showed that the adapted lexicons, when used for polarity classification of tweets, improved accuracy by 1-2%, and F-measure by 1-4% in two datasets over using the original lexicon. In the third dataset, our adapted lexicons gave similar accuracy, but 1.36% lower F-measure than the original lexicon.

Our main conclusion behind this line of work is that the contextual semantics of words, when extracted from tweets and incorporated into lexicon-based approaches, tend to improve sentiment analysis performance over traditional, non-semantic approaches on Twitter.

8.2 CONCEPTUAL SEMANTICS FOR SENTIMENT ANALYSIS OF TWITTER

In Chapter 4 we investigated the second research question of this thesis:

RQ2 ***Could the conceptual semantics of words enhance sentiment analysis performance?***

To this end, we proposed several methods for extracting and using the semantic concepts (e.g. person, company, city), which represent the entities (e.g. Steve Jobs, Vodafone, London) extracted from tweets, as features in both supervised and lexicon-based approaches to increase their performance in tweet-level sentiment analysis.

For supervised sentiment classification, we used Naive Bayes as a case study of supervised classifiers and investigated three methods of incorporating our proposed semantic features into this classifier: by replacement, by augmentation, and by interpolation. We experimented with three Twitter datasets and compared the performance of semantic features against two type of features extracted from the syntactic representation of words. We also compared against sentiment-topic features. Results showed that the semantic features outperformed, on average, both types of syntactic features in positive and negative classification of tweets by 5-6% and 1-4% in F-measure, respectively. Compared to sentiment-topic features, our semantic features improved performance on negative sentiment classification by around 1%, but gave 1.5% lower performance on positive sentiment classification.

For lexicon-based sentiment analysis, we enriched the SentiCircle representation with conceptual semantics using the augmentation method and performed sentiment analysis at tweet-level using the lexicon-based approach built on top of SentiCircle. Our average

evaluation results across three Twitter datasets showed performance improvement of 0.39% in F-measure, compared to using SentiCircle with no semantic enrichment.

Overall, our findings in this line of work demonstrated the high potential of incorporating conceptual semantics as features in Twitter sentiment analysis. Our experimental results suggested that conceptual semantic features outperforms other types of features when the datasets being analysed are large and cover a wide range of topics.

8.3 SEMANTIC PATTERNS FOR SENTIMENT ANALYSIS OF TWITTER

While in Chapters 3 and 4 we investigated using the contextual and conceptual semantics of individual words in sentiment analysis, in Chapter 5 we proposed an approach for extracting patterns of words of similar contextual semantics and sentiment in tweets (SS-Patterns) and using these patterns to improve sentiment analysis performance. Our aim behind this work was to address the third research question of this thesis:

RQ3 ***Could semantic sentiment patterns boost sentiment analysis performance?***

We experimented with our approach on 9 Twitter datasets and validated the extracted patterns by using them as classification features in entity- and tweet-level sentiment analysis tasks. To this end, we trained several supervised classifiers from our patterns and compared their performance against models trained from 6 state-of-the-art syntactic and semantic type of features. Evaluation results showed that our patterns consistently outperformed all other sets of features in both, entity- and tweet-level sentiment analysis tasks. At the tweet level, SS-Patterns improved the classification performance by 1.94% in accuracy and 2.19% in F-measure on average. Also, at the entity level, our patterns produced 2.88% and 2.64% higher accuracy and F-measure than all other features respectively.

We also conducted a quantitative and qualitative analysis on a sample of our extracted patterns, and showed that our patterns are strongly consistent with the sentiment of words within them. Also, our analysis showed that our approach was able to find patterns of entities indicating sentiment controversy or disagreement in tweets.

Our findings in Chapter 5 suggest that contextual semantic and sentiment similarities of words tend to exist in tweets, by means of certain patterns. These patterns, when used

as features to train sentiment classifiers, outperform other types of features extracted from the syntactic or semantic representations of words.

8.4 ANALYSIS ON STOPWORD REMOVAL METHODS FOR SENTIMENT ANALYSIS OF TWITTER

In Chapter 6 we investigated the fourth and last research question of this thesis:

RQ4 ***What effect does stopword removal have on sentiment classification performance?***

Our goal from addressing the above question was to explore how words of low informativeness values or weak semantics (aka stopwords) in a given Twitter corpus impact sentiment analysis performance. To achieve this goal, we applied six stopword removal methods to tweets from six Twitter datasets and analysed how removing stopwords affects two supervised sentiment classification methods, Maximum Entropy (MaxEnt) and Naive Bayes (NB). We assessed the impact of removing stopwords by observing fluctuations on: (i) the level of data sparsity, (ii) the size of the classifier's feature space and (iii), the classifier's performance in terms of accuracy and F-measure.

Our results indicated that removing stopwords based on using general stopword lists lowered the sentiment classification performance by 1.04% in accuracy, and 1.24% in and F-measure. Nevertheless, our literature survey showed that this removal method was widely used in previous works (see Section 6.1).

On the other hand, our results showed that identifying and removing stopwords based on supervised learning methods produced the highest sentiment classification performance, but gave a moderate reduction on the feature space and had no impact on the data sparsity. Lastly, our results showed that removing singleton words is the method that obtains the best trade-off, reducing the feature space and data sparsity degree substantially, while maintaining a high classification performance.

Part V

APPENDIX

A

EVALUATION DATASETS FOR TWITTER SENTIMENT ANALYSIS

We present 9 Twitter datasets that we have used in different parts of this thesis. Our choice of using these specific datasets in our analysis was because they are: (i) publicly available to the research community, (ii) manually annotated, providing a reliable set of judgements over the tweets and, (iii) used to evaluate several sentiment sentiment analysis models. Tweets in these datasets have been annotated with different sentiment labels including: *Negative*, *Neutral*, *Positive*, *Mixed*, *Other* and *Irrelevant*. Table 31 displays the distribution of tweets in the eight selected datasets according to these sentiment labels.

Variations of the evaluation datasets are due to the particularities of the different sentiment analysis tasks. Sentiment analysis on Twitter, as discussed in Chapter 2, spans multiple tasks, such as polarity detection (positive vs. negative), subjectivity detection (polar vs. neutral) or sentiment strength detection. These tasks can also be performed either at tweet level or at target (entity) level. In the following subsections, we provide an overview of the available evaluation datasets and the different sentiment tasks for which they are used.

Dataset	No. of Tweets	#Negative	#Neutral	#Positive	#Mixed	#Other	#Irrelevant
STS-Test	498	177	139	182	-	-	-
STS-Expand	1000	527	-	473	-	-	-
HCR	2,516	1,381	470	541	-	45	79
OMD	3,238	1,196	-	710	245	1,087	-
SS-Twitter	4,242	1,037	1,953	1,252	-	-	-
Sanders	5,513	654	2,503	570	-	-	1,786
GASP	12,771	5,235	6,268	1,050	-	218	-
WAB	13,340	2,580	3,707	2,915	-	420	3,718
SemEval	13,975	2,186	6,440	5,349	-	-	-

Table 31: Total number of tweets and the tweet sentiment distribution in all datasets.

Stanford Twitter Sentiment Test Set (STS-Test)

The Stanford Twitter sentiment corpus (http://help.sentiment140.com/), introduced by Go et al. [60] consists of two different sets, training and test. The training set contains 1.6 million tweets automatically labelled as positive or negative based on emotions. For example, a tweet is labelled as positive if it contains :), :-), :), :D, or =) and is labelled as negative if it contains :(, :-(, or : (.

The test set (STS-Test), on the other hand, is manually annotated and contains 177 negative, 182 positive and 139 neutrals tweets. These tweets were collected by searching Twitter API with specific queries including names of products, companies and people. Although the STS-Test dataset is relatively small, it has been widely used in the literature in different evaluation tasks. For example, Go et al. [60], Saif et al. [135, 137], Speriosu et al. [152], and Bakliwal et al. [10] use it to evaluate their models for polarity classification (positive vs. negative). In addition to polarity classification, Marquez et al. [22] use this dataset for evaluating subjectivity classification (neutral vs. polar).

Stanford Extended Dataset (STS-Expand)

We built and used this dataset in our evaluation in Chapter 4. It consists of training and test sets. The training set contains 60,000 tweets randomly selected from the original Stanford Twitter Sentiment corpus (STS) [60]. Half of the tweets in this dataset contains positive emoticons, such as :), :-), :), :D, and =), and the other half contains negative emoticons such as :(, :-(, or : (.

Due to the relatively small size of the STS-Test dataset, as shown in the previous section, we proposed extending it by adding 641 tweets randomly selected from the original STS corpus, and annotated manually by 12 users (researchers in our lab), where each tweet was annotated by one user.

The final STS-Expand test set, as shown in Table 31, consists of 527 negatively, and 473 positively annotated tweets.

Health Care Reform (HCR)

The Health Care Reform (HCR) dataset was built by crawling tweets containing the hashtag "#hcr" (health care reform) in March 2010 [152]. A subset of this corpus was manually annotated by the authors with 5 labels (*positive, negative, neutral, irrelevant, unsure(other)*) and split into training (839 tweets), development (838 tweets) and test (839 tweets) sets. The authors also assigned sentiment labels to 8 different targets extracted

from all the three sets (*Health Care Reform, Obama, Democrats, Republicans, Tea Party, Conservatives, Liberals, and Stupak*). However, both the tweet and the targets within it, were assigned the same sentiment label, as can be found in the published version of this dataset (https://bitbucket.org/speriosu/updown). Altogether, the three subsets (training, development and test), as shown in Table 31, consist of 2,516 tweets including 1,381 negative, 470 neutral and 541 positive tweets.

The HCR dataset has been used to evaluate polarity classification [152, 136] but can also be used to evaluate subjectivity classification since it identifies neutral tweets.

Obama-McCain Debate (OMD)

The Obama-McCain Debate (OMD) dataset was constructed from 3,238 tweets crawled during the first U.S. presidential TV debate in September 2008 [144]. Sentiment labels were acquired for these tweets using Amazon Mechanical Turk, where each tweet was rated by at least three annotators as either *positive, negative, mixed*, or *other*. The authors in [47] reported an inter-annotator agreement of 0.655, which shows a relatively good agreement between annotators. The dataset is provided at https://bitbucket.org/speriosu/updown along with the annotators' votes on each tweet. We considered those sentiment labels, which more than half of the voters agree on, as final labels of the tweets. This resulted in a set of 1,196 negative, 710 positive and 245 mixed tweets.

The OMD dataset is a popular dataset, which has been used to evaluate various supervised learning methods [73, 152, 136], as well as unsupervised methods [72] for polarity classification of tweets. Tweets' sentiments in this dataset were also used to characterize the Obama-McCain debate event in 2008 [47].

Sentiment Strength Twitter Dataset (SS-Tweet)

This dataset consists of 4,242 tweets manually labelled with their positive and negative sentiment strengths. i.e., a negative strength is a number between -1 (not negative) and -5 (extremely negative). Similarly, a positive strength is a number between 1 (not positive) and 5 (extremely positive). The dataset was constructed by [160] to evaluate SentiStrenth (http://sentistrength.wlv.ac.uk/), a lexicon-based method for sentiment strength detection.

In our work in Chapter 5, we re-annotated tweets in this dataset with sentiment labels (negative, positive, neutral) rather than sentiment strengths, which will allow using this dataset for subjectivity classification. To this end, we assign a single sentiment label to

each tweet based on the following two rules inspired by the way SentiStrength works:[1] (i) a tweet is considered neutral if the absolute value of the tweet's negative to positive strength ratio is equals to 1, (ii) a tweet is positive if its positive sentiment strength is 1.5 times higher than the negative one, and negative otherwise. The final dataset, as shown in table 31, consists of 1,037 negative, 1,953 neutral and 1,252 positive tweets.

The original dataset is publicly available at http://sentistrength.wlv.ac.uk/documentation/ along with other 5 datasets from different social media platforms including MySpace, Digg, BBC forum, Runners World forum, and YouTube.

Sanders Twitter Dataset

The Sanders dataset consists of 5,512 tweets on four different topics (Apple, Google, Microsoft, Twitter). Each tweet was manually labelled by one annotator as either *positive, negative, neutral*, or *irrelevant* with respect to the topic. The annotation process resulted in 654 negative, 2,503 neutral, 570 positive and 1,786 irrelevant tweets.

The dataset has been used in [22, 91, 46] for polarity and subjectivity classification of tweets.

The Sanders dataset is available at http://www.sananalytics.com/lab

The Dialogue Earth Twitter Corpus

The Dialogue Earth Twitter corpus consists of three subsets of tweets. The first two sets (WA, WB) contain 4,490 and 8,850 tweets respectively about the weather, while the third set (GASP) contains 12,770 tweets about gas prices. These datasets were constructed as a part of the Dialogue Earth Project[2] (www.dialogueearth.org) and were hand labelled by several annotators with five labels: *positive, negative, neutral, not related* and *can't tell (other)*. In this thesis we merged the two sets about the weather in one dataset (WAB) for our analysis study in Chapter 5. This resulted in 13,340 tweets with 2,580 negative, 3,707 neutral, and 2,915 positive tweets. The GASP dataset on the other hand consists of 5,235 negative, 6,268 neutral and 1,050 positive tweets.

The WAB and the GASP datasets have been used to evaluate several machine learning classifiers (e.g., Naive Bayes, SVM, KNN) for polarity classification of tweets [5].

1 http://sentistrength.wlv.ac.uk/documentation/SentiStrengthJavaManual.doc

2 Dialogue Earth, is former program of the Institute on the Environment at the University of Minnesota.

SemEval-2013 Dataset (SemEval)

This dataset was constructed for the Twitter sentiment analysis task (Task 2) [104] in the Semantic Evaluation of Systems challenge (SemEval-2013).[3] The original SemEval dataset consists of 20K tweets split into training, development and test sets. All the tweets were manually annotated by 5 Amazon Mechanical Turk workers with negative, positive and neutral labels. The turkers were also asked to annotate expressions within the tweets as subjective or objective. Using a list of the dataset's tweet ids provided by [104], we managed to retrieve 13,975 tweets with 2,186 negative, 6,440 neutrals and 5,349 positives tweets.

Participants in the SemEval-2013 Task 2 used this dataset to evaluate their systems for expression-level subjectivity detection[101, 36], as well as tweet-level sentiment detection[96, 127].

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

B

ANNOTATION BOOKLET FOR THE STS-GOLD DATASET

We need to manually annotate 3000 tweets with their sentiment label (Negative, Positive, Neutral, Mixed) using the online annotation tool "Tweenator.com". The task consists of two subtasks:

Task A. Tweet-Level Sentiment Annotation Given a tweet message, decide weather it has a positive, negative, neutral or mixed sentiment.

Task B. Entity-Level Sentiment Annotation Given a tweet message and a named entity, decided weather the entity received a negative, positive or neutral sentiment. The named entities to annotate are highlighted in yellow within the tweets.

Please note that:

- A Tweet could have a different sentiment from an entity within it. For example, the tweet "`iPhone 5 is very nice phone, but I can't upgrade :(`" has a negative sentiment. However, the entity "`iPhone 5`" receives a positive sentiment.
- "Mixed" label refers to a tweet that has mixed sentiment. For example, the "`Kobe is the best in the world not Lebron`" has a mixed sentiment.
- Some tweets might have emoticons such as :), :-), :(, or :-(. Please give less attention to the emoticons and focus more on the content of the tweets. Emoticons can be very misleading indicators sometimes.
- Try to be objective with your judgement and feel free to take a break whenever you feel tired or bored.

BIBLIOGRAPHY

[1] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[2] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41, Athens, Greece, 2009.

[3] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993. ISSN 0163-5808. doi: 10.1145/170036.170072. URL http://doi.acm.org/10.1145/170036.170072.

[4] Fotis Aisopos, George Papadakis, and Theodora Varvarigou. Sentiment analysis of social media content using n-gram graphs. In *Proceedings of the 3rd ACM SIGMM international workshop on Social media*, pages 9–14, Scottsdale, Arizona, USA, 2011. ACM.

[5] Amir Asiaee T, Mariano Tepper, Arindam Banerjee, and Guillermo Sapiro. If you are happy and you know it... tweet. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1602–1606, Maui, Hawaii, USA, 2012. ACM.

[6] Anthony Aue and Michael Gamon. Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP)*, Borovets, Bulgaria, 2005.

[7] Hakan Ayral and Sırma Yavuz. An automated domain specific stop word generation method for natural language text classification. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, pages 500–503, Istanbul, Turkey, 2011. IEEE.

[8] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, Valletta, Malta, 2010.

[9] Younggue Bae and Hongchul Lee. Sentiment analysis of twitter audiences: Measuring the positive or negative influence of popular twitterers. *Journal of the American Society for Information Science and Technology*, 63(12):2521–2535, 2012.

[10] Akshat Bakliwal, Piyush Arora, Senthil Madhappan, Nikhil Kapre, Mukesh Singh, and Vasudeva Varma. Mining sentiments from tweets. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '12, pages 11–18, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2392963.2392970.

[11] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of COLING*, Beijing, China, 2010.

[12] Siddharth Batra and Deepak Rao. Entity based sentiment analysis on twitter. *Science*, 9(4):1–12, 2010.

[13] Adam Bermingham and Alan F Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1833–1836, Toronto, ON, Canada, 2010. ACM.

[14] S.I. Bhuiyan. Social media and its effectiveness in the political reform movement in egypt. *Middle East Media Educator*, 1(1):14–20, 2011.

[15] Celeste Biever. Twitter mood maps reveal emotional states of america. *New Scientist*, 207(2771):14, 2010.

[16] A. Bifet and E. Frank. Sentiment knowledge discovery in twitter streaming data. In *Discovery Science*, Canberra, Australia, 2010.

[17] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, San Francisco, California, 2001. ACM.

[18] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[19] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[20] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[21] Margaret M Bradley and Peter J Lang. 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, 1999.

[22] Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. Combining strengths, emotions and polarities for boosting twitter sentiment analysis. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 2, Chicago, Illinois, USA, 2013. ACM.

[23] Samuel Brody and Nicholas Diakopoulos. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–570, Edinburgh, United Kingdom, 2011. Association for Computational Linguistics.

[24] Erik Cambria. An introduction to concept-level sentiment analysis. In *Advances in Soft Computing and Its Applications*, pages 478–483. Springer, 2013.

[25] Erik Cambria and Amir Hussain. *Sentic computing: Techniques, tools, and applications*, volume 2. Springer, 2012.

[26] Erik Cambria and Bebo White. Jumping nlp curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2):48–57, 2014.

[27] Erik Cambria, Praphul Chandra, Avinash Sharma, and Amir Hussain. Do not feel the trolls. *Proceedings of the 9th ISWC*, 2010.

[28] Erik Cambria, Robert Speer, Catherine Havasi, and Amir Hussain. Senticnet: A publicly available semantic resource for opinion mining. In *AAAI fall symposium: commonsense knowledge*, volume 10, page 02, 2010.

[29] Erik Cambria, Tim Benson, Chris Eckl, and Amir Hussain. Sentic proms: Application of sentic computing to the development of a novel unified framework for measuring health-care quality. *Expert Systems with Applications*, 39(12):10533–10543, 2012.

[30] Erik Cambria, Marco Grassi, Amir Hussain, and Catherine Havasi. Sentic computing for social media marketing. *Multimedia tools and applications*, 59(2):557–577, 2012.

[31] Erik Cambria, Catherine Havasi, and Amir Hussain. Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In *FLAIRS Conference*, pages 202–207, Marco Island, Florida, USA, 2012.

[32] Erik Cambria, Andrew Livingstone, and Amir Hussain. The hourglass of emotions. In *Cognitive Behavioural Systems*, pages 144–157. Springer, 2012.

[33] Erik Cambria, Bjorn Schuller, Yunqing Xia, and Catherine Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, page 1, 2013.

[34] Erik Cambria, Yangqiu Song, Haixun Wang, and Newton Howard. Semantic multidimensional scaling for open-domain sentiment analysis. 2013.

[35] Jaime Guillermo Carbonell. Subjective understanding: Computer models of belief systems. Technical report, DTIC Document, 1979.

[36] Tawunrat Chalothorn and Jeremy Ellman. Tjp: Using twitter to analyze the polarity of contexts. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).*, Atlanta, Georgia, USA, 2013.

[37] David Chandler. Introduction to modern statistical mechanics. *Introduction to Modern Statistical Mechanics, by David Chandler, pp. 288. Foreword by David Chandler. Oxford University Press, Sep 1987. ISBN-10: 0195042778. ISBN-13: 9780195042771*, 1, 1987.

[38] Praphul Chandra, Erik Cambria, and Alvin Pradeep. Enriching social communication through semantics and sentics. *Sentiment Analysis where AI meets Psychology (SAAIP)*, page 68, 2011.

[39] Yejin Choi and Claire Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, Hawaii, 2008. Association for Computational Linguistics.

[40] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[41] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[42] D. A. Cruse. *Meaning in Language: An Introduction to Semantics and Pragmatics*. Oxford University Press, 2004.

[43] Danica Damljanovic and Kalina Bontcheva. Named entity disambiguation using linked data. In *Proceedings of the 9th Extended Semantic Web Conference*, Creet, Greece, 2012.

[44] Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528, Budapest, Hungary, 2003. ACM.

[45] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 241–249, Beijing, China, 2010. Association for Computational Linguistics.

[46] William Deitrick and Wei Hu. Mutually enhancing community detection and sentiment analysis on twitter networks. *Journal of Data Analysis and Information Processing*, 1:19–29, 2013.

[47] N.A. Diakopoulos and D.A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proc. 28th Int. Conf. on Human factors in computing systems*, Atlanta, Georgia, USA, 2010. ACM.

[48] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135, Philadelphia, PA, USA, 2006. ACM.

[49] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the international conference on Web search and web data mining*, pages 231–240, Palo Alto, California, USA, 2008. ACM.

[50] Peter Sheridan Dodds and Christopher M Danforth. Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. *Journal of Happiness Studies*, 11(4):441–456, 2010.

[51] Paul Ekman. Universals and cultural differences in facial expressions of emotion. In *Nebraska symposium on motivation*. University of Nebraska Press, 1971.

[52] John R. Firth. A synopsis of linguistic theory. *Studies in Linguistic Analysis*, 1930-1955.

[53] George Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.

[54] Christopher Fox. Information retrieval data structures and algorithms. *Lexical Analysis and Stoplists*, pages 102–130, 1992.

[55] Aldo Gangemi, Valentina Presutti, and D Reforgiato Recupero. Frame-based detection of opinion holders and topics: A model and a tool. *Computational Intelligence Magazine, IEEE*, 9(1):20–30, 2014.

[56] Lisette Garcia-Moya, Henry Anaya-Sanchez, and Rafael Berlanga-Llavori. Retrieving product features and opinions from customer reviews. *IEEE Intelligent Systems*, 28(3):19–27, 2013. ISSN 1541-1672.

[57] Gizem Gezici, Rahim Dehkharghani, Berrin Yanikoglu, Dilek Tapucu, and Yucel Saygin. Su-sentilab: A classification system for sentiment analysis in twitter. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, 2013.

[58] George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5, 2008.

[59] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N.A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, DTIC Document, 2010.

[60] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009.

[61] Balakrishnan Gokulakrishnan, Pavalanathan Priyanthan, Thiruchittampalam Ragavan, Nadarajah Prasath, and AShehan Perera. Opinion mining and sentiment analysis on a twitter data stream. In *Advances in ICT for Emerging Regions (ICTer), 2012 International Conference on*, pages 182–188, Colombo, Sri Lanka, 2012. IEEE.

[62] Pollyanna Gonçalves, Matheus Araújo, Fabrício Benevenuto, and Meeyoung Cha. Comparing and combining sentiment analysis methods. In *Proceedings of the first ACM conference on Online social networks*, pages 27–38, Boston, Massachusetts, USA, 2013. ACM.

[63] Pollyanna Gonçalves, Fabrício Benevenuto, and Meeyoung Cha. Panas-t: A pychometric scale for measuring sentiments on twitter. *arXiv preprint arXiv:1308.1857*, 2013.

[64] Viktor Hangya, Gábor Berend, and Richárd Farkas. Szte-nlp: Sentiment detection on twitter messages. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).*, Atlanta, Georgia, USA, 2013.

[65] Zellig S Harris. Distributional structure. *Word*, 1954.

[66] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[67] Vasileios Hatzivassiloglou and Kathleen R McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181, Madrid, Spain, 1997. Association for Computational Linguistics.

[68] Catherine Havasi, Robert Speer, and Jason Alonso. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, pages 27–29, Borovets, Bulgaria, 2007.

[69] Huang He. Sentiment analysis of sina weibo based on semantic sentiment space model. In *Management Science and Engineering (ICMSE), 2013 International Conference on*, pages 206–211, Harbin, China, 2013. IEEE.

[70] Yulan He, Chenghua Lin, Wei Gao, and Kam-Fai Wong. Tracking sentiment and topic dynamics from social media. In *Sixth International AAAI Conference on Weblogs and Social Media*, Dublin, Ireland, 2012.

[71] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, Seattle, WA, USA, 2004. ACM.

[72] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web*, pages 607–618, Rio de Janeiro, Brazil, 2013. International World Wide Web Conferences Steering Committee.

[73] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 537–546, Rome, Italy, 2013. ACM.

[74] Yuheng Hu, Fei Wang, and Subbarao Kambhampati. Listening to the crowd: automated analysis of events via aggregated twitter sentiment. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2640–2646, Beijing, China, 2013. AAAI Press.

[75] M.M. Hussain and P.N. Howard. the role of digital media. *Journal of Democracy*, 22 (3):35–48, 2011.

[76] Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Micro-blogging as online word of mouth branding. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 3859–3864, Boston, MA, USA, 2009. ACM.

[77] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160, Portland, Oregon, 2011. Association for Computational Linguistics.

[78] Jaap Kamps, MJ Marx, Robert J Mokken, and Maarten De Rijke. Using wordnet to measure semantic orientations of adjectives. 2004.

[79] Nadin Kökciyan, Arda Celebi, Arzucan Ozgür, and Suzan Usküdarlı. Bounce: Sentiment classification in twitter using rich feature sets. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, 2013.

[80] Efstratios Kontopoulos, Christos Berberidis, Theologos Dergiades, and Nick Bassiliades. Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, 2013.

[81] Sotiris B Kotsiantis, ID Zaharakis, and PE Pintelas. *Supervised machine learning: A review of classification techniques*. 2007.

[82] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the ICWSM*, Barcelona, Spain, 2011.

[83] Adam DI Kramer. An unobtrusive behavioral model of gross national happiness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 287–290. ACM, 2010.

[84] Adam DI Kramer. The spread of emotion via facebook. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 767–770, Austin, Texas, USA, 2012. ACM.

[85] K. Krippendorff. Content analysis: an introduction to its methodology., 1980.

[86] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[87] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384, Hong Kong, China, 2009. ACM.

[88] Chenghua Lin, Yulan He, Richard Everson, and Stefan Ruger. Weakly supervised joint sentiment-topic detection from text. *Knowledge and Data Engineering, IEEE Transactions on*, 24(6):1134–1145, 2012.

[89] Bing Liu. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:568, 2010.

[90] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[91] Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI*, Toronto, Ontario, Canada, 2012.

[92] Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. Automatically building a stopword list for an information retrieval system. In *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*, 2005.

[93] Masoud Makrehchi and Mohamed S Kamel. Automatic extraction of domain-specific stopwords from labeled documents. In *Advances in information retrieval*, pages 222–233. Springer, 2008.

[94] Nikolaos Malandrakis, Abe Kazemzadeh, Alexandros Potamianos, and Shrikanth Narayanan. Sail: A hybrid approach to sentiment analysis. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).*, Atlanta, Georgia, USA, 2013.

[95] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[96] E Martınez-Cámara, A Montejo-Ráez, MT Martın-Valdivia, and LA Urena-López. Sinai: Machine learning and emotion of the crowd for sentiment analysis in microblogs. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, 2013.

[97] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment classification using word sub-sequences and dependency sub-trees. In *Advances in Knowledge Discovery and Data Mining*, pages 301–311. Springer, 2005.

[98] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[99] Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[100] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 2012.

[101] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).*, Atlanta, Georgia, USA, 2013.

[102] Arturo Montejo-Ráez, Eugenio Martínez-Cámara, M Teresa Martín-Valdivia, and L Alfonso Ureña-López. A knowledge-based approach for polarity classification in twitter. *Journal of the Association for Information Science and Technology*, 65(2):414–425, 2014.

[103] A. Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the European Conference on Machine Learning*, pages 318–329, Berlin, Germany, 2006.

[104] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter. In *In Proceedings of the 7th International Workshop on Semantic Evaluation. Association for Computational Linguistics.*, Atlanta, Georgia, USA, 2013.

[105] Sapna Negi, MSD2080 Msida, and Mike Rosner. Uom: Using explicit semantic analysis for classifying sentiments. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, 2013.

[106] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*, 2011.

[107] John C Norcross, Edward Guadagnoli, and James O Prochaska. Factor structure of the profile of mood states (poms): two partial replications. *Journal of Clinical Psychology*, 40(5):1270–1277, 1984.

[108] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. From tweets to polls: Linking text sentiment to public opinion time series. volume 11, pages 122–129, Washington, DC, US, 2010.

[109] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390, 2013.

[110] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC 2010*, Valletta, Malta, 2010.

[111] Georgios Paltoglou and Mike Thelwall. Twitter, myspace, digg: Unsupervised sentiment analysis in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):66, 2012.

[112] Georgios Paltoglou, Mike Thelwall, and Kevan Buckley. Online textual communications annotated with grades of emotion strength. In *Proceedings of the 3rd International Workshop of Emotion: Corpora for research on Emotion and Affect*, pages 25–31, Valletta, Malta, 2010.

[113] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271, Barcelona, Spain, 2004. Association for Computational Linguistics.

[114] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

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

[116] Ravi Parikh and Matin Movassate. Sentiment analysis of user-generated twitter updates using various classification techniques. *CS224N Final Report*, 2009.

[117] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, page 71, 2001.

[118] James W Pennebaker, Matthias R Mehl, and Kate G Niederhoffer. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577, 2003.

[119] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100, Beijing, China, 2008. ACM.

[120] R. Plutchik. *Emotion: Theory, Research, and Experience*. Number v. 4. Acad. Press, 1989. URL http://books.google.co.uk/books?id=qK1ZewAACAAJ.

[121] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. Knowledge extraction based on discourse representation theory and linguistic frames. In *Knowledge Engineering and Knowledge Management*, pages 114–129. Springer, 2012.

[122] Thomas Proisl, Paul Greiner, Stefan Evert, and Besim Kabashi. Klue: Simple and robust methods for polarity classification. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).*, Atlanta, Georgia, USA, 2013.

[123] Matthew Purver and Stuart Battersby. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491, Avignon, France, 2012. Association for Computational Linguistics.

[124] Tushar Rao and Saket Srivastava. Twitter sentiment analysis: How to hedge your bets in the stock markets. In *State of the Art Applications of Social Network Analysis*, pages 227–247. Springer, 2014.

[125] Jonathon Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

[126] Diego Reforgiato Recupero, Valentina Presutti, Sergio Consoli, Aldo Gangemi, and Andrea Giovanni Nuzzolese. Sentilo: frame-based sentiment analysis. *Cognitive Computation*, pages 1–15, 2014.

[127] Robert Remus. Asvuniofleipzig: Sentiment analysis in twitter using data-driven machine learning techniques. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).*, Atlanta, Georgia, USA, 2013.

[128] K Revathy and B Sathiyabhama. A hybrid approach for supervised twitter sentiment classification. *International Journal of Computer Science and Business Informatics*, 7, 2013.

[129] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

[130] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proc. the 2003 conference on Empirical methods in natural language processing*, Sapporo, Japan, 2003.

[131] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, United Kingdom, 2011. Association for Computational Linguistics.

[132] G. Rizzo and R. Troncy. Nerd: Evaluating named entity recognition tools in the web of data. In *Workshop on Web Scale Knowledge Extraction (WEKEX'11)*, volume 21, Bonn, Germany, 2011.

[133] Carlos Rodrıguez-Penagos, Jordi Atserias, Joan Codina-Filba, David Garcıa-Narbona, Jens Grivolla, Patrik Lambert, and Roser Saurı. Fbm: Combining lexicon-based ml and heuristics for social media polarities. In *In Proceedings of the seventh*

international workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, Georgia, USA, 2013.

[134] Magnus Sahlgren. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1): 33–54, 2008.

[135] H. Saif, Y. He, and H. Alani. Semantic Smoothing for Twitter Sentiment Analysis. In *Proceeding of the 10th International Semantic Web Conference (ISWC)*, Bonn, Germany, 2011.

[136] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In *Proceedings of the 11th international conference on The Semantic Web*, Boston, MA, USA, 2012.

[137] Hassan Saif, Yulan He, and Harith Alani. Alleviating data sparsity for twitter sentiment analysis. In *Proceedings, 2nd Workshop on Making Sense of Microposts (#MSM2012) in conjunction with WWW 2012*, Layon, France, 2012.

[138] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Evaluation datasets for twitter sentiment analysis a survey and a new dataset, the sts-gold. In *Proceedings, 1st Workshop on Emotion and Sentiment in Social and Expressive Media (ESSEM) in conjunction with AI*IA Conference*, Turin, Italy, 2013.

[139] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In *Proc. 11th Extended Semantic Web Conf. (ESWC)*, Crete, Greece, 2014.

[140] Barry S Sapolsky*, Daniel M Shafer, and Barbara K Kaye. Rating offensive words in three television program contexts. *Mass Communication and Society*, 14(1):45–70, 2010.

[141] Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

[142] FW Scholz. Maximum likelihood estimation. *Encyclopedia of statistical sciences*, 1985.

[143] Yohei Seki, David Kirk Evans, Lun-Wei Ku, Hsin-Hsi Chen, Noriko Kando, and Chin-Yew Lin. Overview of opinion analysis pilot task at ntcir-6. In *Proceedings of NTCIR-6 Workshop Meeting*, pages 265–278, Tokyo, Japan, 2007.

[144] D.A. Shamma, L. Kennedy, and E.F. Churchill. Tweet the debates: understanding community annotation of uncollected sources. In *Proceedings of the first SIGMM workshop on Social media*, pages 3–10, Beijing, China, 2009. ACM.

[145] Catarina Silva and Bernardete Ribeiro. The importance of stop word removal on recall values in text categorization. In *Proceedings of the International Joint Conference on Neural Networks*, volume 3, pages 1661–1666, Portland, Oregon, USA, 2003. IEEE.

[146] Ismael Santana Silva, Janaína Gomide, Adriano Veloso, Wagner Meira Jr, and Renato Ferreira. Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 475–484, Beijing, China, 2011. ACM.

[147] Tomer Simon, Avishay Goldberg, Limor Aharonson-Daniel, Dmitry Leykin, and Bruria Adini. Twitter in the cross fire-the use of social media in the westgate mall terror attack in kenya. *PloS one*, 9(8):e104136, 2014.

[148] Mark P. Sinka and David W. Corne. Design and application of hybrid intelligent systems. pages 1015–1023, Amsterdam, The Netherlands, The Netherlands, 2003. IOS Press. ISBN 1-58603-394-8. URL http://dl.acm.org/citation.cfm?id=998038.998149.

[149] Mark P Sinka and David W Corne. Towards modernised and web-specific stoplists for web document analysis. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, Beijing, China, 2003. IEEE.

[150] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2330–2336, Barcelona, Catalonia, Spain, 2011. AAAI Press.

[151] David Sontag and Dan Roy. Complexity of inference in latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1008–1016, Granada, Spain, 2011.

[152] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the EMNLP First workshop on Unsupervised Learning in NLP*, Edinburgh, Scotland, 2011.

[153] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

[154] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560, Fortaleza, Ceara, Brazil, 2008. ACM.

[155] Carlo Strapparava and Alessandro Valitutti. Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086, Lisbon, Portugal, 2004.

[156] Jared Suttles and Nancy Ide. Distant supervision for emotion classification with discrete binary values. In *Computational Linguistics and Intelligent Text Processing*, pages 121–136. Springer, 2013.

[157] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 133–140, Ann Arbor, Michigan, USA, 2005. Association for Computational Linguistics.

[158] Mike Thelwall and David Wilkinson. Public dialogs in social network sites: What is their purpose? *Journal of the American Society for Information Science and Technology*, 61(2):392–404, 2010.

[159] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[160] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012.

[161] Tun Thura Thet, Jin-Cheon Na, Christopher SG Khoo, and Subbaraj Shakthikumar. Sentiment analysis of movie reviews on discussion boards using a linguistic approach. In *Proc. the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, Hong Kong, China, 2009.

[162] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. *Urbana*, 51:61801.

[163] Cherie Courseault Trumbach and Dinah Payne. Identifying synonymous concepts in preparation for technology mining. *Journal of Information Science*, 33(6):660–677, 2007.

[164] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. volume 10, pages 178–185, Washington, DC, US, 2010.

[165] P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, Pennsylvania, 2002.

[166] Peter Turney and Michael Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21: 315–346, 2003.

[167] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[168] Marieke Van Erp, Giuseppe Rizzo, and Raphaël Troncy. Learning with the web: Spotting named entities on the intersection of nerd and machine learning. In *The 3rd International Workshop on Making Sense of Microposts (#MSM'13), Concept Extraction Challenge*, pages 27–30, Rio de Janeiro, BRAZIL, 2013. Citeseer.

[169] William N Venables and Brian D Ripley. *Modern applied statistics with S*. Springer, 2002.

[170] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120, Jeju Island, Korea, 2012. Association for Computational Linguistics.

[171] Lan Wang and Yuan Wan. Sentiment classification of documents based on latent semantic analysis. In *Advanced Research on Computer Education, Simulation and Modeling*, pages 356–361. Springer, 2011.

[172] David Watson, Lee A Clark, and Auke Tellegen. Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, 54(6):1063, 1988.

[173] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631, Bremen, Germany, 2005. ACM.

[174] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.

[175] Olof Wijksgatan and Lenz Furrer. Gu-mlt-lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).*, Atlanta, Georgia, USA, 2013.

[176] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics.

[177] Ludwig Wittgenstein. *Philosophical Investigations*. Blackwell, London, UK, 1953, 2001.

[178] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM, 2012.

[179] Tao Xu, Qinke Peng, and Yinzhao Cheng. Identifying the semantic orientation of terms using s-hal for sentiment analysis. *Knowledge-Based Systems*, 35:279–289, 2012.

[180] Yan Xu, Gareth JF Jones, JinTao Li, Bin Wang, and ChunMing Sun. A study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems*, 3(3):1007–1012, 2007.

[181] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 177–186, Hong Kong, China, 2011. ACM.

[182] Yiming Yang. Noise reduction in a statistical approach to text categorization. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and*

development in information retrieval, pages 256–263, Seattle, Washington, USA, 1995. ACM.

[183] Omar Zaidan, Jason Eisner, and Christine D Piatko. Using" annotator rationales" to improve machine learning for text categorization. In *HLT-NAACL*, pages 260–267, Rochester, NY, USA, 2007.

[184] Ley Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. Combining lexiconbased and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89, 2011.

[185] Lumin Zhang, Yan Jia, Bin Zhou, and Yi Han. Microblogging sentiment analysis using emotional vector. In *Second International Conference on Cloud and Green Computing (CGC)*, pages 430–433, Xiangtan,China, 2012. IEEE.

[186] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 56–65. Association for Computational Linguistics, 2010.

[187] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

[188] George Kingsley Zipf. Human behavior and the principle of least effort. 1949.