UNIVERSITEIT
GENT

FACULTEIT LETTEREN
EN WIJSBEGEERTE

Can machines sense irony?

Exploring automatic irony detection on social media

Automatische ironiedetectie op sociale media

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Taalkunde aan de Universiteit Gent te verdedigen door

Cynthia Van Hee

Dit onderzoek werd gefinancierd door het
AMiCA IWT SBO-project 120007.

Gent, 2017

Promotoren:
Prof. dr. Véronique Hoste
Prof. dr. Els Lefever

to my parents

to Bert

Abstract

The development of the social web has stimulated creative language use like irony. As a result, research in automatic irony detection has thrived in the past few years, to improve our understanding of ironic language on the one hand, and to enhance text mining applications that suffer from irony (e.g. automatic sentiment analysis) on the other. In this thesis, we present a comprehensive approach to modelling irony, including the development of a new fine-grained annotation scheme, a varied set of experiments to detect irony automatically, and an extrinsic evaluation of the irony detection system by means of a sentiment analysis use case. An important contribution of this research includes a new approach to model implicit or prototypical sentiment, which is crucial in irony detection.

We assembled a gold-standard corpus of English tweets using irony-related hashtags (i.e. *#irony, #sarcasm, #not*), which was manually annotated according to a new annotation scheme. The scheme is grounded in irony literature and provides for a fine-grained annotation, including the identification of different forms of irony and the specific text spans that realise the irony in a tweet. This manually annotated dataset allowed us to investigate two things: the linguistic realisation of irony in online text, and the viability of our machine learning approach to irony detection.

Analysis of the annotated corpus analysis revealed that one in five instances in the corpus are not ironic despite containing an irony hashtag, which confirms that manual annotations are instrumental for this task. We also observed that in 70% of the ironic tweets, a polarity contrast takes place between the literal and the implied message (i.e. *ironic by clash*), whereas situational irony and *other* irony only represent 30% of the ironic tweets.

Experiments using a support vector machine showed that irony detection benefits from a variety of information sources. While related research is often based on lexical features, we found that syntactic and semantic features are important predictors for irony as well. Combining the three information sources outperformed a strong character *n*-gram baseline and yielded a state-of-the-art F_1-score of 70.11% showing a good balance between precision and recall.

A qualitative analysis revealed that polarity contrasts which include implicit sentiment (e.g. 'I love going to the dentist') present a critical bottleneck in irony detection. An important contribution of this thesis is that we took the first steps to automatically define the prototypical sentiment related to particular situations (e.g. 'going to the dentist'). We compared a knowledge base (SenticNet) and data driven (Twitter) approach and found that the latter allows to infer prototypical sentiment with high accuracy ($> 70\%$). Using such prototypical sentiment information to inform the irony detection system increased recall of the latter considerably ($> 83\%$).

In the last part of this thesis, we present a sentiment analysis use case as the extrinsic evaluation of our irony detection system. We investigated the effect of irony on automatic sentiment analysis at the tweet level and observed that the latter shows an important drop in performance when applied to ironic text. By informing the sentiment analysis system with the output of our irony detection system, we were able to augment its performance considerably (20% to 40%).

Can machines sense irony? We found they show at least good performance by using a support vector machine exploiting a varied feature set. Informing the classifier with implicit or prototypical sentiment further enhances its performance and present promising directions for future research.

Samenvatting

Steeds meer communicatie verloopt via sociale media, die bijgevolg vaker gekenmerkt worden door figuurlijk en creatief taalgebruik. Een voorbeeld van zulk creatief taalgebruik is ironie, een stijlfiguur die de jongste jaren vaak onderzocht wordt om een beter inzicht te krijgen in de manieren waarop we communiceren, maar ook om onderzoek naar automatische tekstanalyse (bv. automatische sentimentanalyse) te verbeteren. In dit proefschrift beschrijven we een uitgebreide aanpak om ironie te modelleren, inclusief een theoretisch kader met nieuwe annotatierichtlijnen voor ironie, experimenten om ironie automatisch te detecteren en een *use case* als extrinsieke evaluatie van het detectiesysteem. Een belangrijke bijdrage van dit proefschrift zijn onze experimenten om automatisch impliciet of prototypisch sentiment te herkennen, wat een belangrijke rol speelt in ironisch taalgebruik.

We stelden een corpus samen van Engelstalige tweets met behulp van ironie-gerelateerde hashtags (i.e. *#irony, #sarcasm, #not*). Al deze tweets werden manueel geannoteerd aan de hand van een nieuw annotatieschema. Het schema is gebaseerd op literatuur over ironie en laat niet alleen toe om aan te duiden of een tweet al dan niet ironisch is, maar stelt ook een meer fijnmazige aanpak voor waarbij het type ironie kan aangeduid worden. Dit manueel geannoteerde corpus liet ons toe om te onderzoeken i) wat de tekstuele kenmerken zijn van ironie in dit type data en ii) hoe goed een lerend systeem in staat is ironie te detecteren.

Een analyse van het geannoteerde corpus toonde aan dat één op de vijf tweets

in het corpus niet ironisch is, hoewel een ironie-gerelateerde hashtag aanwezig is. Dit toont aan dat manuele annotaties noodzakelijk zijn voor deze taak. De analyse toonde verder aan dat in 70% van de ironische tweets een ander sentiment wordt uitgedrukt dan eigenlijk wordt bedoeld (i.e. iets positiefs zeggen om een negatief sentiment of negative opinie uit te drukken); de resterende 30% bevat voorbeelden van situationele en andere vormen van ironie.

Experimenten met een *support vector machine* algoritme toonden aan dat ironiedetectie baat heeft bij een gevarieerde groep informatiebronnen of *features*. Hoewel vergelijkbare studies vaak gebruik maken van lexicale features, stelden we vast dat syntactische en semantische features ook belangrijke indicatoren zijn voor ironie op Twitter. Met een combinatie van de drie bovenvermelde featuregroepen behaalt het systeem een betere score dan de *baseline* (i.e. een referentiescore van een doorgaans eenvoudiger systeem) met een F_1-score van 70.11%.

Een kwalitatieve analyse toonde aan dat ironische tweets die impliciet sentiment bevatten (bv. 'Joepie, straks naar de tandarts!') vaak moeilijker te detecteren zijn. Daarom hebben we in dit proefschrift de eerste stappen gezet om het impliciet sentiment van bepaalde situaties (bv. 'naar de tandarts gaan') automatisch te detecteren. We vergeleken twee aanpakken (SenticNet en Twitter) en toonden aan dat de Twittergebaseerde aanpak een goede methode is om impliciet sentiment automatisch te bepalen (i.e. accuraatheid van 70%). Door informatie over impliciet sentiment toe te voegen aan het systeem voor ironiedetectie verbetert de performantie van het systeem duidelijk (> 88%).

In het laatste deel van dit proefschrift beschrijven we een *use case* als extrinsieke evaluatie van het detectiesysteem voor ironie. We onderzochten het effect van ironie op automatische sentimentanalyse en toonden aan dat de performantie van dat laatste sterk vermindert als ironie aanwezig is in de tweet. Bijgevolg stelden we vast dat, als het systeem voor sentimentanalyse geïnformeerd wordt door automatische ironiedetectie, de performantie opnieuw sterk verbetert (+ 20% tot 40%).

Kunnen computers ironie herkennen? We tonen in dit proefschrift aan dat ze in ieder geval goede accuraatheden behalen met behulp van *support vector machines* en een gevarieerde set *features*. Door informatie over impliciet of prototypisch sentiment toe te voegen, verbetert de performantie nog verder.

Acknowledgements

As a colleague and friend would have it, “you don’t write a thesis on your own”. I am happy to look back at the past four years, which have been an enjoyable and challenging experience, and to thank the people who have supported me in one way or another during this period.

First of all, I would like to thank my supervisor, Prof. Dr. Véronique Hoste and copromotor Prof. Dr. Els Lefever, who have been instrumental to this work. Véronique, thank you for giving me the opportunity to work in the stimulating environment LT3 is. Thank you for your trust and ambitious goals, which have made me achieve things that I would not have imagined possible a few years ago. Els, thank you for your help with many struggles, and for your ability to make the biggest obstacles seem surmountable. Your optimism and kindness are exceptional. Thank you both for your scientific input and warm personality.

To Prof. Dr. Walter Daelemans, I want to express my sincere gratitude for being in my thesis committee and jury. Thank you Walter, for your critical insights and support, for giving me the opportunity to collaborate on the AMiCA project, and for, on occasion, offering me an office ‘across the water’. I am also grateful to Dr. Alexandra Balahur, Dr. Iris Hendrickx, Prof. Dr. Bernard De Clerck and Dr. Orphée De Clercq, who kindly agreed to be in my jury.

I warmly thank Gitte for doing the cover layout of this thesis and Kristien for proofreading the text.

Over the past four years, I have had the opportunity to work with wonderful colleagues, not only in Ghent. Thank you Walter, Ben, Guy, Chris, Janneke and many other colleagues, for the productive AMiCA cooperation, which allowed me to take my first steps into research. Thank you Anneleen, for trusting me as your internship co-supervisor. Thank you Frederic, Laura and Olivier, for bravely annotating thousands of tweets, and for the interesting discussions when annotating irony and (not so) common sense. I am also grateful to all department colleagues, for interesting research talks and joyous lunch conversations.

A special thank you goes to the colleagues of LT3, whose talent and contagious enthusiasm turn our offices into a stimulating and enjoyable workplace. Thank you Peter, Nils and Gilles, for sharing an office with me and for your sense of humor, especially on Friday afternoon, reminding us that the weekend was just around the corner. Thank you Lieve, for assisting me in my first teaching experience. Thank you Arda, Ayla, Bram, Claudia, Joke, Julie, Laura and Stef, for being such kind and helpful colleagues livening up many lunch breaks, hackathons and teambuildings. Thank you Orphée and Bart, for teaching me so much and for sharing experiences, from guiding students to writing theses and even building homes. You have been more than 'just' colleagues during these past months, and your encouragements have been crucial to finishing this thesis on time. Of course, thank you Véronique and Els. A big thank you is also addressed to some former colleagues and office mates. Marjan, thank you for the fruitful and pleasant collaboration on sentiment analysis, and Sarah, for your (much needed) help with coding and the enjoyable office talks in the morning. Thank you Isabelle, Klaar and Mariya for being there, always kind and caring.

A heartfelt thank you goes to my dearest friends, who have been there for many years: Annelies, Lisa and Heleen. I am grateful for so many joyful moments together and look forward to sharing many more. Thank you for being such strong and loving people, and for sharing good and bad. Thank you Leen, Selome, Ellen and Tine, for your support and friendship in the past years, and for the joyful talks, even when it has been a while.

My final words of gratitude are reserved for my family. Kristine, Luc and Bert, thank you for being such a wonderful family-in-law, for your support and the delicious meals on Wednesdays. Thank you Dennis, for being the brother I never had, for your contagious enthusiasm and big heart. Kelsey, thank you for being such a wonderful sister and friend, and the most passionate teacher I have known. I am so proud of what you already have and still will achieve. Thank you, mama en papa, for your unconditional love and support, and for raising us to be independent and loving people. Thank you for always being there when we need you, and for teaching us to never give up.

Bert, thank you for your tremendous love and support -especially in the past months, for letting me share my drama and staying calm when I was not. Thank you, for sharing so much happiness and for being the most important part of my life.

Contents

1 Introduction **1**

1.1 Background and research motivation 2

1.2 Research objectives 5

1.3 Thesis outline 6

2 Related research **9**

2.1 Defining irony 11

2.2 Verbal irony versus sarcasm 14

2.3 Computational approaches to verbal irony 15

3 Resources **21**

3.1 Corpora 21

3.2 Annotation 23

3.2.1 Annotation scheme 23

3.2.2 Inter-annotator agreement 27

3.2.3 Corpus analysis 29

3.3 Summary 33

4 Automatic irony detection 35

4.1 Introduction 35

4.2 Experimental corpus 36

4.3 Preprocessing and feature engineering 37

4.3.1 Lexical features 39

4.3.2 Syntactic features 41

4.3.3 Sentiment lexicon features 41

4.3.4 Semantic features 42

4.3.5 Feature statistics 43

4.4 Experimental setup 44

4.4.1 Machine learning method 44

4.4.2 Classifier evaluation 45

4.5 Baselines and experimental results 46

4.5.1 Baseline classifiers 46

4.5.2 Experimental results 47

4.5.3 Analysis 54

4.6 Summary 57

5 Modelling connotative knowledge for irony detection 59

5.1 Introduction 60

5.2 Approach 63

5.2.1 Using SenticNet to infer implicit sentiment 65

5.2.2 Crawling Twitter to infer implicit sentiment 70

5.3 Summary 82

6 Automatic irony detection based on a polarity contrast 85

6.1 Introduction 85

6.2 A polarity clash-based approach 89

6.2.1 Gold-standard implicit sentiment 89

6.2.2 Automatic implicit sentiment 92

6.3 Combining an SVM with polarity contrast information 94

6.3.1 SVM exploiting a polarity contrast feature 94

6.3.2 A hybrid system 97

6.4 Summary 99

7 Using irony detection for sentiment analysis: a use case 101

7.1 Automatic sentiment analysis 102

7.1.1 Background 102

7.1.2 System description 103

7.1.3 Experimental setup and model optimisation 107

7.1.4 Results 108

7.2 Irony-sensitive sentiment analysis 109

7.2.1 Dataset 110

7.2.2 Experimental setup 110

7.2.3 Results 111

7.3 Summary 114

8 Conclusion 117

8.1 Annotating irony in social media text 118

8.2 Detecting irony in social media text 119

8.3 Modelling implicit sentiment 122

8.4 Sentiment analysis use case 125

A Publications 143

B Guidelines for annotating irony in social media text 147

References 162

CHAPTER 1

Introduction

The development of the social web has stimulated the use of figurative and creative language, including irony, in public. From a philosophical/psychological perspective, discerning the mechanisms that underlie ironic speech improves our understanding of human reasoning and communication, but more and more, this interest in understanding irony emerges from the machine learning community (Wallace 2015). In fact, the frequent use of irony on social media has important implications for natural language processing (NLP) tasks, which aim to understand and produce human language (Turing 1950). Although various definitions of irony co-exist, it is often identified as a trope or figurative language use whose actual meaning is different from what is literally enunciated. As such, modelling irony has a large potential for applications in various research areas, including text mining, author profiling, detecting online harassment and, perhaps one of the most investigated applications at present, automatic sentiment analysis.

State-of-the-art systems for irony detection mostly rely on bag-of-words and syntactic information like part-of-speech tags. However, the use of such 'shallow' information for a subjective task like irony detection has been questioned (Wallace 2015). Moreover, to facilitate data collection and annotation, many supervised learning approaches rely on hashtag-labelled Twitter data, although it has been shown to increase data noise (Kunneman et al. 2015). In this thesis, we aim to model irony in social media data by combining lexical and syntactic information with sentiment and semantic features. To minimise the noise in our dataset, all tweets are manually annotated based on a set of newly developed coding principles to mark irony in social media text.

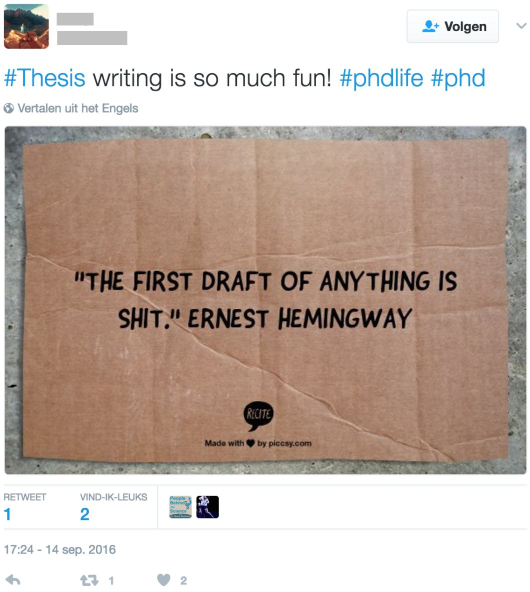
A crucial step before modelling irony is to recognise it and understand how it is linguistically realised. We therefore propose a theoretic framework that benefits automatic irony detection and present one of the first fine-grained annotation schemes for irony in social media text. We explore to what extent irony is susceptible to computational modelling and adopt machine learning techniques to develop a system for automatic irony recognition in social media text. In a next step, we valorise the potential of automatic irony detection to enhance the state of the art in sentiment analysis by means of a use case.

1.1 Background and research motivation

Irony has always played an important role in human communication, although its functions may vary. Vlastos (1987) described it as the instrument of a moral lesson (i.e. 'Socratic irony'), while recent studies mostly agree that it is used to express ridicule or scorn (Wilson and Sperber 1992). In the framework of politeness theory (Brown and Levinson 1987), irony is considered a face-protecting strategy when uttering criticism or refuting someone's idea. More recently, Veale and Hao (2009) revealed that, when used on social media, irony may function as a way to stand out or to express creativity in writing, as illustrated in Figure 1.1.

The extensive use of social media we have witnessed in the past decade has increased researchers' interest in analysing this new type of text to better understand human thoughts and communication. Not only individuals, but more and more companies and organisations have a keen interest in understanding how consumers evaluate their goods and services, knowing that online opinions are likely to influence the decisions made by others. The study of such opinions, attitudes and beliefs is known as *automatic sentiment analysis* or *opinion mining* (Pang and Lee 2008) and has become one of the main research domains in natural language processing at present. State-of-the-art sentiment classifiers have been developed in the context of specialised shared tasks like SemEval (Nakov et al. 2013, Rosenthal et al. 2014, 2015, Nakov et al. 2016, Rosenthal et al. 2017) and have flourished in industry through commercial applications (Liu 2012).

Nevertheless, many applications struggle to maintain high performance when applied to ironic text (Liu 2012, Maynard and Greenwood 2014, Ghosh and



[redacted]

#Thesis writing is so much fun! #phdlife #phd

Vertalen uit het Engels

"THE FIRST DRAFT OF ANYTHING IS SHIT." ERNEST HEMINGWAY

RETWEET 1 VIND-IK-LEUKS 2

17:24 - 14 sep. 2016

Figure 1.1: Example of an ironic tweet.

Veale 2016). Like other types of figurative language, ironic text should not be interpreted in its literal sense; it requires a more complex understanding based on associations with context or world knowledge. As such, modelling ironic text as an example of the complexity of human communication has a large potential for applications in diverse research areas: literary science, language psychology, sociolinguistics, and computational linguistics.

Previous work in the latter field has proven that irony undermines the performance of text analysis tools (Liu 2012) and hereby influences NLP tasks including sentiment analysis, which is illustrated by the following examples:

(1) I love how my mom says she can count on Rion more than me. #not #jealous.

Regular sentiment analysis systems would probably classify example 1 as positive, whereas the intended sentiment is undeniably negative. In this sentence, the hashtag *#not* hints at the presence of irony, but many other ironic utterances, like sentence 2 are devoid of such explicit indications.

(2) I feel so blessed to get ocular migraines.

For human readers, it is clear that the author of sentence 2 does not feel blessed at all, but wants to communicate the opposite. This can be inferred from the contrast between the positive sentiment expression ‘I feel so blessed’, and the negative connotation associated with getting ocular migraines. Although such connotative information is easily understood by most people, it is difficult to access by machines.

Cyberbullying detection would be another text mining application the performance of which may be undermined by the presence of irony. Studies on automatic cyberbullying detection have shown that implicit instances of cyberbullying are often overlooked by the classifier (i.e. *false negatives*) (e.g. Dadvar 2014, Van Hee et al. 2015). Such instances typically lack explicit profane words, and the offense is often made through irony, as shown in example 3.

(3) Go ahead drop me hate, I’m looking forward to it.

In sum, to enhance the performance of similar tasks, and more generally “any model that hopes to make sense of human communication or expression”(Wallace 2015, p. 468), building computational models for detecting irony is of key importance. To be able to do so, it is important to understand how irony is linguistically realised and to identify aspects and forms of irony that are susceptible to computational analysis.

In this thesis, we explore the feasibility of automatic irony detection using machine learning techniques. More specifically, we aim to understand how irony is realised in social media text and look to elaborate a theoretic framework that benefits its automatic detection. To this end, we establish a working definition that is grounded in irony literature and present a new set of annotation guidelines for the textual annotation of irony in an English social media corpus. The scheme allows for a fine-grained annotation below instance level to mark text spans that realise the irony.

In a next step, the manually annotated corpus is used to develop and evaluate an automatic system for irony detection. For this purpose, a series of binary classification experiments are conducted where we explore the predictive power of a varied set of information sources, including lexical, shallow syntactic, sentiment and semantic information. Furthermore, we hypothesise that implicit sentiment (also referred to as *connotative knowledge* or *prototypical sentiment*) information may benefit automatic irony detection. To this end, we explore how it can be derived automatically, starting from manually annotated prototypical sentiment situations (e.g. ‘going to the dentist’), and investigate its conditional added value for irony recognition.

To valorise the potential of accurate irony detection in the domain, a final step consists in an extrinsic evaluation of our classifier. In this use case, we will explore the extent to which automatic irony recognition benefits automatic sentiment analysis in social media text.

The current approach focusses on English Twitter data, but the applied method is language independent, provided that training data are available.

1.2 Research objectives

In accordance with the research motivation in the previous paragraphs, our main research questions can be formulated as follows:

1a. **How is irony realised in social media text like tweets?**

1b. **Can ironic instances be automatically detected in English tweets? If so, which information sources contribute most to classification performance?**

This thesis also aims to provide answers to two related, more specific research questions:

2. Is it feasible to automatically detect implicit or prototypical sentiment related to particular situations and does our approach benefit automatic irony detection?
3. Can our automatic irony detection approach enhance state-of-the-art sentiment classification?

To answer these questions, the following research objectives can be defined:

- **Providing a theoretical framework of irony.**
  To be able to model irony, it is key to understand how it is realised in text and to find out which characteristics are susceptible to computational analysis. An overview of irony literature in linguistics and computational linguistics should provide valuable insights into this complex rhetorical device.
- **Constructing a manually annotated irony dataset.**
  Previous research has shown the potential of Twitter data for training automatic text classification algorithms. However, relying on hashtags as

gold labels to collect ironic tweets has shown to generate data noise and provides little insight into the linguistic realisation of irony on social media. We therefore construct a manually annotated corpus of English ironic tweets, for which we developed one of the first fine-grained annotation schemes for irony.

- **Developing a model for irony detection based on varied NLP features.**
  We explore to what extent automatic irony detection is feasible and investigate the potential of a varied set of features for this task.

- **Developing a method to infer implicit or prototypical sentiment related to particular situations.**
  Related work in irony literature has underlined the importance of implicit or prototypical sentiment information for irony detection. Based on manually annotated prototypical sentiment situations (e.g. 'going to the dentist'), we aim to devise a method to infer such implicit sentiment automatically.

- **Investigating the benefits of automatic irony detection for sentiment classification.**
  Related research has underlined the potential of irony detection to improve sentiment analysis. We therefore evaluate our irony detection system by means of a sentiment analysis use case.

1.3 Thesis outline

This thesis consists of eight chapters, which are structured as follows. **Chapter 2** provides an introduction into irony research, in both linguistics and natural language processing. It presents some of the main theories on irony and discusses related work on machine learning approaches to irony detection. Attention is paid to definitions of irony that have been applied in previous research, as well as data collection methods and feature engineering.

Chapter 3 presents the corpus of English tweets that was created for this research. A great part of this chapter is dedicated to the development of a new fine-grained annotation scheme for irony and discusses how it was validated and applied to our corpus.

Chapter 4 focusses on the irony detection experiments conducted in this thesis. It describes the cleaning and preprocessing of the experimental corpus prior to feature extraction. Next, it presents a set of binary classification experiments using support vector machines (SVM) and exploiting various feature groups combining lexical, sentiment, semantic and syntactic information. A qualitative analysis provides insights into the strenghts and bottlenecks of the approach.

In **Chapter 5**, we take the first steps to modelling implicit (or *prototypical*) sentiment related to a set of concepts and situations. For this purpose, we explore the use of the lexico-semantic knowledge base SenticNet 4, and a data-driven method using Twitter.

Assessing the added value of implicit sentiment information for irony detection is the topic of **Chapter 6**. We apply the techniques described in Chapter 5 to detect prototypical sentiment and evaluate the performance of our irony detection system when provided with information about prototypical sentiment in a tweet.

Chapter 7 describes the use case where we investigate the impact of our irony detection system on an optimised sentiment classifier.

Finally, **Chapter 8** concludes this thesis with our main findings, some limitations of the present research, and perspectives for future work.

CHAPTER 2

Related research

While irony is ubiquitous in human interactions and presents a widely-covered research topic, defining it is an arduous task, and differentiating irony from related terms like *sarcasm* might be even more challenging. Various definitions of irony have been proposed in the literature, and as many have been criticised or refuted. Until today, a uniform definition is still lacking in the field, and the relation between irony and associated concepts is subject to an ongoing debate.

In what follows, we present an overview of irony literature by highlighting seminal work in linguistics and computational linguistics or natural language processing. Based on these insights, we propose a working definition of irony that represents the main guideline for the corpus annotation as detailed in Chapter 3.

The Oxford Dictionary provides three definitions of irony:

- *the expression of one's meaning by using language that normally signifies the opposite, typically for humorous or emphatic effect (synonyms: sarcasm, bitterness, cynicism,...)*;
- *a state of affairs or an event that seems deliberately contrary to what one expects and is often wryly amusing as a result (synonyms: paradox, peculiarity,...)*;

- *a literary technique, originally used in Greek tragedy, by which the full significance of a character's words or actions is clear to the audience or reader although unknown to the character.*

The above definitions refer to what is in irony literature known as **verbal irony**, **situational irony** and **dramatic irony**, respectively. According to the Merriam-Webster dictionary for English, the following should also be considered irony:

- *a pretense of ignorance and of willingness to learn from another assumed in order to make the other's false conceptions conspicuous by adroit questioning –called also Socratic irony.*

Kreuz and Roberts (1993) distinguish four types of irony: i) **Socratic irony** and ii) **dramatic irony**, both explained as a tension between what the hearer knows and what the speaker pretends to know (with the latter entailing a performance aspect), iii) **irony of fate/situational irony**, which involves an incongruence between two situations, and iv) **verbal irony**, which implies a speaker who intentionally says the opposite of what he or she believes.

While Socratic irony and dramatic irony find their origin in Ancient Greek comedy, nowadays, a taxonomy of irony generally consists of situational and verbal irony. Situational irony, or irony of fate as described by Kreuz and Roberts (1993), refers to situations that fail to meet some expectations (Lucariello 1994, Shelley 2001). An example tweet that describes situational irony is presented in example 4.

(4) "The irony is that despite all our crews and help from the MWRA [Massachusetts Water Resource Authority] with all sorts of detection crews, it was a Town Meeting member who discovered the break and reported it to officials." (Shelley 2001, p. 787)

Verbal irony is traditionally identified as a trope or figurative language use where enunciated words imply something other than their principal signification. In other words, their literal meaning has to be inferred through interpretation. As described in Burgers (2010), the classical definition of verbal irony is attributed to the author Quintiliano (1959) and states that verbal irony implies saying the opposite of what is meant. Until today, this traditional account has influenced many conceptualisations of irony, one of the most well-known probably being Grice's in his theory of conversational implicature (1975, 1978). Although this standard definition is commonly used in research on irony detection (Kunneman et al. 2015), it has faced criticism (e.g. Giora 1995, Sperber and Wilson 1981), and a number of adjustments and alternatives to this approach have been proposed.

In what follows, we highlight seminal work in irony literature and describe the state of the art in automatic irony detection. While we discuss the most relevant studies for the present research, we refer to the overview papers by Wallace (2015) and Joshi, Bhattacharyya and Carman (2016) for a comprehensive analysis of linguistic and computational approaches to irony. Important to note is that when discussing related research, we refer to irony using the terminology employed by the corresponding researchers (i.e. 'sarcasm', 'irony' or 'verbal irony').

2.1 Defining irony

In this section, we discuss a number of definitions and theories of irony. We start with the seminal work by Grice (1975), who introduces irony in the framework of his conversational implicature theory. The theory explains four principles (i.e. *conversational maxims*) that govern human communication by defining mutual expectations related to:

- **Quantity**: make your contribution as informative as required, but not more than required;
- **Quality**: make a contribution that is true;
- **Relation**: be relevant;
- **Manner**: be orderly and brief, and avoid obscurity and ambiguity.

Within this framework, Grice (1975) defines irony, like other forms of figurative language, as a violation of the maxim of Quality. By violating one of the maxims, the speaker aims to attract the hearer's attention and to encourage him to 'dig deeper' to understand that irony is being used. As such, the speaker of an ironic utterance implicates some other thing than (i.e. generally the opposite of) what they literally say. To respond to some critiques (e.g. Sperber and Wilson 1981), Grice later adds subjectivity to falsity as a requirement for irony. As such, to be ironic, an utterance is "intimately connected with the expression of a feeling, attitude or evaluation" (1978, p. 53).

Although Grice's theory of conversation (1975) has impacted widely on language philosophy and semantics, his view on verbal irony has been questioned (e.g. Sperber and Wilson 1981, Giora 1995). In what follows, we highlight some critiques towards and alternatives to his approach. The latter come from different directions, including irony as an *echoic mention* (Sperber and Wilson 1981), irony as an *(allusional) pretense* (e.g. Clark and Gerrig 1984, Currie 2006, Kumon-Nakamura et al. 1995), and irony as a form of indirect negation (Giora 1995).

According to Sperber and Wilson (1981), Grice's (1975) account of irony is not necessary (e.g. it does not cover ironic questions or understatements), nor sufficient (i.e. not all utterances that are false are ironic). The researchers state that Grice's theory (1975) fails to explain more subtle variants of irony, including understatements, allusions, and exclamations (examples 5, 6 and 7, respectively).

(5) (When a customer is complaining in a shop, blind with rage)
You can tell he's upset. (Wilson and Sperber 1992, p. 54)

(6) (When said in a rainy rush-hour traffic jam in London)
When a man is tired of London, he is tired of life. (Wilson and Sperber 1992, p. 55)

(7) (After arriving in Tuscany, where it is windy and rainy at that moment)
Ah, Tuscany in May! (Wilson and Sperber 1992, p. 55)

As an alternative, they propose the **Echoic Mention Theory**, involving that in speaking ironically, "a speaker echoes a remark in such a way as to suggest that he finds it untrue, inappropriate, or irrelevant" (Sperber and Wilson 1981, p. 307). According to this theory, ironic statements implicitly allude to a previous (explicit or implicit) proposition, and thereby express the speaker's negative attitude towards it. As such, the irony in examples 6 and 7 targets the speaker's negative attitude towards the hearer's previously uttered claim that London is a fantastic city and that the weather in Tuscany is always beautiful. One may debate, however, the ironic character of example 5, the difference of which compared to a mere understatement is not made clear.

Another post-Gricean approach to verbal irony that is worth mentioning here, is the **Pretense Theory** by among others Clark and Gerrig (1984), Currie (2006) and Kumon-Nakamura et al. (1995). They consider irony *allusive*, implying that the speaker pretends to say something other than they mean to draw the hearer's attention to some failed expectation or norm. Unlike Grice's (1975) approach, the theory claims that irony involves pragmatic rather than semantic insincerity or falsehood. What mainly distinguishes the theory from the Echoic

Mention Theory, is that the latter assumes the presence of an initial utterance, whereas the Pretense Theory does not. As such, the theory would explain ironic utterances where it is hard to infer the hearer's initial utterance, such as insincere compliments (e.g. "You sure know a lot"), questions (e.g. "How old did you say you were?"), and over-polite requests (e.g. "Would you mind very much if I asked you to consider cleaning up your room some time this year?")[1]. Similarly to ironic understatements mentioned by Sperber and Wilson (1981), one may doubt the ironic character of the latter two examples, as the difference compared to (non-ironic) rhetorical questions and exaggerations or hyperboles remains unexplained.

Giora (1995), finally, describes irony as an **indirect negation** strategy. This theory of verbal irony seems to reconcile elements from both the traditional or Gricean approach (i.e. irony implies violation of a norm through meaning inversion) and so-called 'post-Gricean' approaches (i.e. explaining why irony is used, while attenuating the notion of meaning inversion). The researcher describes irony as an indirect negation strategy where a broad interpretation of negation is assumed, including understatements and exaggerations. Giora (1995) also states that, unlike the traditional, pretense, and echoic approaches to irony, the indirect negation theory assumes that the ironic interpretation of an utterance does not replace the literal one, but that both meanings are activated to underline the discrepancy or contrast between them. For instance, when uttering the ironic phrase "what a lovely party", the hearer both understands the literal meaning as the expectation of the speaker, as well as the implied one (i.e. the party is rather boring) as its true opinion. Interestingly, this co-existence of the literal and intended meaning is described as the distinguishing factor between irony and humor, as in the latter only the literal expression, which causes the humorous effect, is understood. In the why of using irony, Giora (1995) (similarly to Brown and Levinson 1987) sees a politeness strategy enabling its users to negate or criticise something in a face-protecting way.

While the above paragraphs present only a small proportion of linguistic approaches to irony, they demonstrate that many theories and different conceptualisations of the phenomenon exist. Both the Gricean and post-Gricean approaches to irony have been widely discussed and alternative approaches have been suggested (e.g. Coulson 2005, Kihara 2005, Ritchie 2005, Utsumi 2000). Burgers (2010) presents a comprehensive overview of different theories on the subject and identifies a number of characteristics that are shared by many irony theories (e.g. its implicit and subjective character, the presence of an opposition between what is said and what is intended). Similarly, Camp (2012) combines crucial elements of the so-called *semantic* and *pragmatic* approaches to sarcasm and defines propositional, lexical, like-prefixed and illocutionary sarcasm as four

[1] Examples by Kumon-Nakamura et al. (1995).

subtypes of the phenomenon that are each based on another interpretation of meaning inversion.

While Grice's (1975) theory has been criticised from different directions, we believe that his approach, if taking into account his note about a necessarily related sentiment expression (Grice 1978), covers a substantial number of ironic instances. In fact, the main criticism towards Grice's approach is that it fails to explain i) more subtle variants of irony, and ii) why irony would be preferred over a sincere utterance. However, as mentioned earlier in this section, many of these critics often fail to provide a clear explanation of such subtler forms of irony (e.g. how ironic hyperboles differ from non-ironic ones). Moreover, although it does not focus on the pragmatics of irony, the theory suggests that it is a form of pretense, a view that is later extended by the Pretense Theory.

Consequently, our working definition of verbal irony (i.e. irony that is realised in text) is based on this traditional approach and describes irony as *an evaluative expression whose polarity (i.e. positive, negative) is inverted between the literal and the intended evaluation, resulting in an incongruence between the literal evaluation and its context.* This definition is comparable to that of Burgers' (2010), since it has shown to cover most written forms of irony as identified in the literature.

2.2 Verbal irony versus sarcasm

When describing how irony works, many studies have also struggled to distinguish between irony, in particular verbal irony, and sarcasm. To date, opinion on the definition of verbal irony and how it relates to sarcasm is very much divided. Some theorists consider sarcasm and irony to be the same or consistently refer to one term without specifying whether and how the two phenomena differ (e.g. Burgers 2010, Clark and Gerrig 1984, Davidov et al. 2010, Grice 1975), whereas others posit that they are significantly different (Haiman 1998, Lee and Katz 1998a) or only partially overlap (e.g. Attardo 2000, Barbieri and Saggion 2014, Clift 1999, Kreuz and Roberts 1993).

According to the differentiating view, sarcasm is a form of verbal irony which has a more aggressive tone (Attardo 2000), is directed at someone or something (Kreuz and Roberts 1993, Sperber and Wilson 1981), and is used intentionally (Barbieri and Saggion 2014, Gibbs et al. 1995, Haiman 1998). Furthermore, sarcasm is often considered as a way to express ridicule (Clift 1999, Joshi, Bhattacharyya and Carman 2016, Lee and Katz 1998a) and negativity (Camp 2012, Clift 1999). Some researchers have also pointed to vocal aspects that differentiate sarcasm and verbal irony, showing that cues such as nasality (Haiman 1998), a slower tempo, a lower pitch level and greater intensity (Rockwell 2000) are significant indicators of sarcasm.

The above-mentioned theories point out a number of differences between verbal irony and sarcasm, such as the level of aggressiveness, the presence of a target, the intention to hurt, and even some vocal clues (e.g. nasality). It is unclear, however, whether these features provide sufficient evidence of a clear-cut distinction between irony and sarcasm, since not all of them are easy to recognise. In fact, among others, Tsur et al. (2010) and Eisterhold et al. (2006) claim that there is no way of formally distinguishing between the terms, and many researchers consequently consider sarcasm and irony as strongly related (Hallmann et al. 2016b). Another reason why researchers do not differentiate between irony and sarcasm is the observation of a shift in meaning between the two terms. Over time, the term 'sarcasm' seems to have gradually replaced what was previously designed by 'irony' (Nunberg 2001). In their experimental study, Bryant and Fox Tree (2002) and (Gibbs 1986) both found evidence for this meaning shift, observing that student respondents understood the term 'sarcasm' better than 'verbal irony'. Moreover, Bryant and Fox Tree revealed that while student respondents were able to identify and define sarcasm, they were "unable to provide a reasonable definition of irony" (2002, p. 15). Consequently, they often considered instances of verbal irony to be sarcastic.

It is clear from the above paragraphs that, while research efforts on irony and sarcasm are expanding, a formal definition of both phenomena is still lacking in the literature. As a response to this ongoing debate, most computational approaches on this subject do not distinguish between either. Indeed, Kunneman et al. (2015) employ the term 'sarcasm' although 'verbal irony' would be the more appropriate term in some cases, and Filatova uses both terms to refer to the same phenomenon, stating that "it is not possible to create a definition of irony or sarcasm to identify ironic utterances following a set of formal criteria" (2012, p. 392). For these reasons, we do not distinguish between the terms either, and we will consistently use the term 'irony' throughout this thesis.

2.3 Computational approaches to verbal irony

Analysing subjective text has attracted a great deal of research interest in the past decade. As the amount of opinionated data has grown thanks to social media platforms like Facebook and Twitter, so has research on text mining. As a result, the past years have witnessed important advances in the field of sentiment analysis. Nevertheless, being trained on 'regular' texts (i.e. the majority of which is non-figurative), such systems suffer from decreased performance when applied to figurative language like irony. As a result, research in natural language processing (NLP) has seen various attempts to tackle this problem by exploring automatic irony detection. Although comparison between different approaches is hard due to a number of variables (e.g. corpus size, definition of irony, class distribution, evaluation method), we present a brief overview of the state of the art in Table 2.1.

Research	Corpus	Balanced?	Approach	Features	Results
Davidov et al. (2010)	Amazon (5.5K), Twitter (1.5K),	✗	SASI	punctuation, syntactic patterns	F=0.83, F=0.55
González-Ibáñez et al. (2011)	Twitter (2.7K)	✓	SMO,	n-grams, LIWC matches, punctuation, emoticons, ToUser	Acc=0.65
Reyes et al. (2013)	Twitter (40K)	✓	Naïve Bayes, Decision Tree	style, emot. scenarios, signatures, unexpectedness	F=0.73
Riloff et al. (2013)	Twitter (3K)	✗	SVM (RBF) + lexicon-based	n-grams, polarity contrast	F= 0.51
Barbieri and Saggion (2014)	Twitter (40K)	✓	Decision Tree	frequency, style, structure, intensity, synonyms, ambiguity, sentiments	F=0.74
Kunneman et al. (2015)	Twitter (812K)	✓	Balanced Winnow	word n-grams	AUC= 0.85, Recall= 0.87
Bouazizi and Ohtsuki (2016)	Twitter (9K)	✓	Random Forrest	lexical, sentiment, syntactic, pattern-based	F_1= 0.81%
Ghosh and Veale (2016)	Twitter (39K)	✓	SVM, Neural Networks	n-grams, PoS, CNN,LSTM, DNN	F_1=0.66, F_1=0.92
Van Hee et al. (2016b)	Twitter (4.8K)	✓	SVM	lexical, syntactic, sentiment, semantic	F_1=0.68
Poria et al. (2016)	Twitter (100K)	✓	CNN-SVM	Word2Vec, sentiment, emotion, personality	F_1= 0.77

Table 2.1: State-of-the-art approaches to irony detection.

As described by Joshi, Bhattacharyya and Carman (2016), recent approaches to irony can roughly be classified into rule-based and (supervised and unsupervised) machine learning-based methods. While rule-based approaches mostly rely upon lexical information and require no training, machine learning invariably makes use of training data and exploits different types of information sources (or *features*), including bags of words, syntactic patterns, sentiment information or semantic relatedness. Recently, deep learning techniques have gained increasing popularity for this task as they allow to integrate semantic relatedness by making use of, for instance, word embeddings.

Regardless of the method used, irony detection often involves a binary classification task where irony is defined as instances that express the opposite of what is meant (e.g. Bouazizi and Ohtsuki 2015, Joshi, Bhattacharyya and Carman 2016, Riloff et al. 2013). Twitter has been a popular data genre for this task, as it is easily accessible and contains self-describing hashtags like *#irony* and *#sarcasm* which allow to collect much data in a rapid way. Moreover, when used as class labels, such hashtags can reduce manual annotation efforts, although this is often at the cost of annotation quality (see Chapter 3). While most approaches have been focussing on English data, irony detection has also been investigated in other languages, including Italian (Barbieri et al. 2014), French (Karoui et al. 2017), Czech (Ptáček et al. 2014), Portuguese (Carvalho et al. 2009) and Dutch (Kunneman et al. 2015). Van Hee et al. (2016c) have been the first to construct a fine-grained annotated dataset of English and Dutch ironic tweets.

Early studies on irony detection include the work of Davidov et al. (2010) and González-Ibáñez et al. (2011). Davidov et al. (2010) focussed on tweets and Amazon product reviews and made use of the semi-supervised algorithm SASI (Tsur et al. 2010) exploiting punctuation information and syntactic patterns. Their sarcasm classifier was trained on manually annotated Amazon and Twitter data and obtained F-scores of respectively 0.79 and 0.83 on a held-out evaluation set. It is not entirely clear, however, whether sarcasm hashtags were removed from the data prior to training. Similarly, Bouazizi and Ohtsuki (2016) made use of part-of-speech tags to extract patterns from the training data that characterise sarcastic and non-sarcastic tweets. In addition to sentiment, lexical and syntactic features, they extracted more than 300,000 patterns from the (extended) training data. By combining all features, their sarcasm classifier yielded an F_1-score of 0.81.

Reyes et al. (2013) focussed on distinguishing ironic tweets from tweets about education, politics, and humorous tweets. The ironic data were collected with the *#irony* hashtags, while the non-ironic corpus was created using the hashtags *#education, #humor* and *#politics.* They introduced different feature types exploiting lexical (e.g. punctuation marks, emoticons, character n-grams, polarity n-grams), syntactic (e.g. contrasting verb tenses) and semantic information (semantic similarity and the relation to emotional contexts such as *pleasantness*).

Performing pairwise binary classification experiments, their approach yielded F-scores of up to 0.76. Barbieri and Saggion (2014) conducted a similar experiment using the same dataset and a wide variety of features (i.e. word frequency, written versus spoken style, adjective/adverb intensity, synonym use, degree of ambiguity, sentence length, punctuation/emoticon use, and sentiments). Compared with Reyes et al. (2013), they achieved slightly better results for distinguishing irony from the education and politics topics, but not for the humour topic.

Kunneman et al. (2015) collected Dutch tweets with sarcasm-related hashtags (e.g. *#sarcasme, #ironie, #not, #cynisme*) and trained a classifier by contrasting the tweets against a background corpus devoid of such hashtags. Their system obtained an AUC-score of 0.85 and recall of 0.87 by making use of word n-gram features. The researchers demonstrated, however, that sarcasm recognition is a hard task in an open setting; manual inspection of the tweets that were classified as sarcastic in the background corpus (i.e. without sarcasm-related hashtags) revealed that 35% of the top-250 ranked tweets were indeed sarcastic. This demonstrates that i) evidently, not all sarcastic tweets are marked with a hashtag, and ii) sarcasm is realised in different ways on Twitter. Riloff et al. (2013) worked with a manually-labelled corpus and applied a hybrid approach combining a supervised SVM exploiting n-gram features with a rule-based *contrast approach*. Suggesting that sarcasm emerges from a contrast between a positive sentiment phrase and a negative situation phrase, the researchers created lists of seed terms for both categories (e.g. 'love' and 'being ignored'), which were expanded through bootstrapping. The experimental results revealed that the approaches are complementary as the contrast method identified ironic tweets that were overlooked by the SVM classifier. Also using manually annotated data, Van Hee et al. (2016c) developed a pipeline extracting four types of features for irony detection in English tweets, including lexical (n-grams, punctuation, capitalisation), syntactic (named entity and part-of-speech information), sentiment (number of positive, negative sentiment words + tweet polarity score), and semantic (distributional cluster information based on Word2Vec word embeddings) information sources. By means of binary classification experiments using SVM, they showed that combining all feature groups benefits classification performance, reaching an F_1-score of 0.68.

In line with Wallace's (2015) claim that text-based features are too shallow and that context and semantics are required for reliable irony detection, deep learning techniques introducing semantic information have recently gained popularity. A recent study by Ghosh and Veale (2016) describes sarcasm detection using neural networks. The researchers compared the performance of an SVM model exploiting lexical features based on word frequency, syntactic features based on part-of-speech information and sentiment features to that of a combined neural network model exploiting word embedding information. They demonstrated that the latter outperformed the SVM-model (F= 0.73), yielding an F-score of 0.92 when hashtag information (e.g. *#sarcasm*) was included in the data. Joshi, Tripathi, Patel, Bhattacharyya and Carman (2016) expanded their set of lexical (e.g. unigrams, quotation marks, laughter expressions) and sentiment (positive/negative sentiment words, LIWC categories[2]) features with word embedding information. They made use of hashtag-labelled book reviews as training data and obtained an F-score of 0.80 using Word2Vec to construct word embeddings. Similarly, Van Hee et al. (2016b) made use of Word2Vec word embeddings to create semantic clusters from a large background corpus containing ironic and non-ironic text, and showed that the features achieve similar performance to lexical features exploiting bags of words, while not relying on the training data. Finally, Poria et al. (2016) made use of deep learning techniques for sarcasm detection on Twitter. Their convolutional neural network (CNN) includes local features from pretrained CNNs that provide information about sentiment, emotion and personality, and combined them with Word2Vec features initialised using a large background corpus and extended using the training data. By feeding the resulting feature vectors to an SVM-classifier, their approach yielded an F_1-score of 0.77.

As mentioned earlier, irony detection has a large potential for natural language processing tasks like sentiment analysis. Bouazizi and Ohtsuki (2015) and Maynard and Greenwood (2014) demonstrated its importance for sentiment analysis by showing performance increases between 3% and 50% when the system is informed about irony presence. The SemEval-2015 task on ‘Sentiment Analysis of Figurative Language in Twitter’ incited researchers to develop a system that correctly determines the sentiment expressed in figurative content (i.e. containing irony, sarcasm and metaphor) (Ghosh et al. 2015). The training data consisting of merely figurative (mostly ironic) tweets, however, the results on the test data showed that most participating systems either performed well on ironic tweets or on metaphorical or regular (i.e. non-ironic) tweets, but not on both (e.g. Van Hee, Lefever and Hoste 2015).

The above paragraphs provide insights into related work on irony detection. It is noteworthy, however, that many of the discussed papers make use of much larger training corpora (up to 812K tweets), whereas the current corpus is limited to 5K tweets (see Chapter 3). Moreover, in the above-described studies (except Davidov et al. 2010, Maynard and Greenwood 2014, Riloff et al. 2013, Van Hee et al. 2016b), training data is often obtained by collecting tweets using the hashtags *#sarcasm*, *#irony* and *#not* and labelling them accordingly (i.e. tweets containing such a hashtag are labelled as ironic, while others are considered non-ironic). In fact, Joshi, Bhattacharyya and Carman (2016) state

[2] Pennebaker et al. (2001)

that most Twitter-based approaches to irony detection make use of hashtag-labelled corpora. An important contribution of the present research is that, after collecting data based on irony-related hashtags, all tweets were manually labelled based on a newly developed annotation scheme for irony (Van Hee et al. 2016a). Manual annotations were preferred to hashtag labelling for several reasons. First, manual annotations limit noise in the corpus caused by hashtag labelling (Kunneman et al. 2015, Van Hee et al. 2016c). Second, the development of a fine-grained annotation scheme allowed to distinguish different forms of irony (see Chapter 3 for details on the annotation process), and hence provided insights into the realisation of irony on social media. Third, and most importantly, during the annotation process, annotators indicated text spans that realise polarity contrasts in ironic tweets, providing us with valuable information about the use of explicit and implicit sentiment expressions in ironic tweets.

CHAPTER 3

Resources

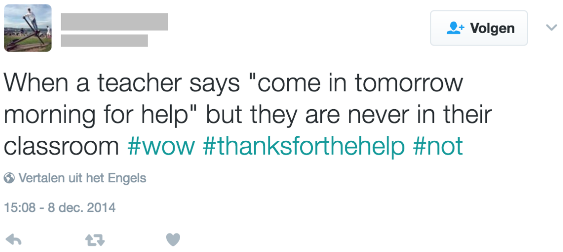
In the previous chapters, we introduced the challenges of automatic irony detection and found that a manually annotated dataset is instrumental to the task. For this research, ironic data were collected using Twitter, a widely used microblogging service and a popular genre for similar tasks.

In this chapter, we aim to answer the first part of our main research question, namely **'how is irony realised in social media text?'**. We describe the construction and annotation of an English Twitter corpus and we introduce a new fine-grained annotation scheme for irony on social media. Next, we discuss the results of an inter-annotator agreement study to assess the validity of our guidelines and we conclude the chapter with a detailed corpus analysis.

3.1 Corpora

To be able to train an irony detection system, a large set of irony examples is necessary. In this section, we describe the construction of an English corpus of tweets and the development of fine-grained annotation guidelines for irony.

To operationalise the task of irony detection, we constructed a dataset of 3,000 English tweets. Since ironic tweets are far less common than regular tweets, we searched the social network for the hashtags *#irony, #sarcasm* and *#not*. For this purpose, we made use of Tweepy[1], a Python library to access the official Twitter API, which provides programmatic access to read Twitter data. This way, we collected approximately 15,000 tweets between 01/12/2014 and 04/01/2015, 3,000 of which were randomly selected as our corpus and manually annotated. The tweets have an average length of 15 tokens and represent 2,676 unique Twitter users. An example tweet is presented in Figure 3.1.



[redacted]

When a teacher says "come in tomorrow morning for help" but they are never in their classroom #wow #thanksforthehelp #not

Vertalen uit het Engels

15:08 - 8 dec. 2014

Figure 3.1: Corpus example.

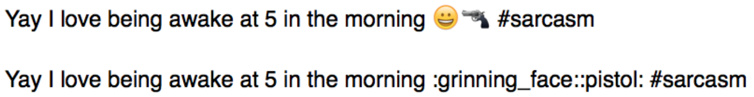
Using hashtags as class labels could, however, affect the quality of the dataset. In fact, Kunneman et al. (2015) demonstrated that hashtags used as gold labels introduce approximately 10% noise into the dataset. To overcome this problem and hence minimise the noise in our corpus, all tweets were manually labelled using a fine-grained annotation scheme for irony. This way, tweets whose irony-related hashtag was considered groundless given the restricted context of the tweet itself, could be identified. Given the absence of fine-grained coding principles for this task, we developed a new annotation scheme that is described in Section 3.2.1.

Prior to data annotation, the entire corpus was cleaned by removing retweets, duplicates and non-English tweets, as well as handling slash- and XML-escaped characters (e.g. &amp;). For practical reasons related to data annotation, all emoji were replaced by their name or description using the Python emoji module[2], which provides the entire set of emoji codes as defined by the unicode consortium, in addition to a number of aliases. An example of this replacement is shown in Figure 3.2.

After cleaning the corpus, we proceeded to its annotation, which is explained in the following sections.

[1] https://github.com/tweepy/tweepy

[2] https://pypi.python.org/pypi/emoji



Yay I love being awake at 5 in the morning 😀🔫 #sarcasm

Yay I love being awake at 5 in the morning :grinning_face::pistol: #sarcasm

Figure 3.2: Example of the emoji replacement.

3.2 Annotation

Prior to building computational models for recognising irony, it is key to understand how irony is linguistically realised and to identify characteristics of the phenomenon that could be modelled. Moreover, a supervised machine learning approach requires labelled training data. It has been demonstrated (*supra*) that hashtag labels are insufficient for accurate irony detection. Hence, we propose manual annotation based on a set of newly developed coding principles. The guidelines are described in great detail by Van Hee et al. (2016c) and present a methodology to annotate irony in social media text. They can be consulted in Appendix B. To assess the validity of our guidelines, an inter-annotator agreement study was carried out in two rounds, each with time different annotators. Finally, this section presents some corpus statistics based on the annotations (see Section 3.2.3).

3.2.1 Annotation scheme

Compared to computational approaches to irony detection, corpus-based linguistic studies on irony are rather scarce. Recently, a number of annotation schemes have been developed (e.g. Bosco et al. 2013, Riloff et al. 2013, Stranisci et al. 2016), although most of them describe a binary distinction (i.e. ironic versus not ironic) without distinguishing between different types of irony or combine its annotation with that of sentiment and opinions.

Nevertheless, to be able to understand how irony is realised in text, a more fine-grained annotation is required. In this section, we describe the construction of a fine-grained annotation scheme for irony in social media text. The scheme allows to distinguish between different types of irony and to indicate the text spans that realise the irony, in order to understand how irony is realised in text. As far as we know, only Karoui et al. (2017) have done similar work.

Literature shows that irony is often realised by means of a polarity contrast (see Chapter 2). As the starting point of the annotation process, we therefore define irony as an *evaluative expression whose polarity (i.e. positive, negative) is* *inverted between the literal and the intended evaluation, resulting in an incongruity between the literal evaluation and its context* (Van Hee et al. 2016a). Such evaluations can be explicit (i.e. *evaluative expressions*), or implicit (i.e. irony *targets*). The latter are text spans that contain no subjective words, but implicitly convey a positive or negative sentiment (e.g. 'I love not being able to sleep!').

The guidelines therefore describe *ironic by means of a polarity clash* as a form of **verbal** irony (i.e. realised in text) that arises from two contrasting sentiment expressions. The scheme further distinguishes the categories *other type irony* and *not ironic.* While the latter is meant for instances that are clearly not ironic, the former can be used for instances that do not contain a polarity contrast, but that are nevertheless considered ironic. This category is further divided into *situational irony* (i.e. situations where the outcome is opposite to the expectations) and *(other) verbal irony.* The three main annotation categories we distinguish are presented below.

- **Ironic by means of a polarity clash:** in accordance with our definition, the text expresses an evaluation whose literal polarity is opposite to the intended polarity.
- **Other type of verbal irony:** there is no contrast between the literal and the intended evaluation, but the text is still ironic. Within this category, a further distinction is drawn between instances describing **situational** irony and other forms of **verbal** irony.
- **Not ironic:** the text is not ironic.

In case of irony resulting from a polarity clash or contrast, the annotators made two supplementary annotations to gain insight into the linguistic realisation of this type of irony. Firstly, they indicated the **harshness** of an instance on a two-point scale (i.e. zero meaning that the tweet is not harsh, one indicating that it is), indicating to what extent the irony is meant to ridicule or hurt someone. The intuition underlying this annotation is grounded in irony literature, stating that harshness could be a distinguishing factor between irony and sarcasm (e.g. Attardo 2000, Clift 1999). Example 8 presents such a harsh tweet.

(8) Thanks mom for all those lovely words, you just love to let me know how proud you are of me #not #wordshurt

Secondly, the annotators also indicated whether an irony-related hashtag (e.g. *#sarcasm, #irony, #not*) was required to recognise the irony, as is the case in example 9. As opposed to example 8, the tweet is not considered harsh.

(9) This should be fun next spring. #not

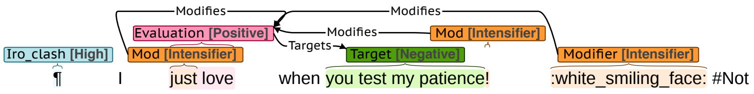
In short, at the tweet level, annotators indicated whether an instance was ironic (either by means of a polarity contrast or by another type of irony) or not. Next and below tweet level, the annotators marked:

- **Evaluative expressions:** text spans (e.g. verb phrases, predicative expressions, emoticons) that express an explicit evaluation. Additionally, a polarity (i.e. positive or negative) had to be indicated for each evaluative expression.
- **Modifiers**: (if present) words that alter the prior polarity of the evaluation (e.g. ‘unbelievably thoughtful’).
- **Targets**: text spans whose implicit sentiment (i.e. connotation) contrasts with that of the literally expressed evaluation.

We are aware that identifying such targets and attributing an implicit sentiment to them is not trivial, this is also why an extensive inter-annotator agreement experiment was set up. In fact, defining whether a concept evokes a positive or negative sentiment is subjective and may vary because of cultural or personal differences. As such, ‘winter weather’ may for instance have a positive connotation for someone who is fond of the holidays, skiing, Christmas, and so on, whereas it might evoke a negative sentiment for people who relate it to extreme cold, rain, icy roads, etc. In the same way, being touched during conversation can evoke annoyance in some people, while it is accepted by or natural to others. Bearing in mind that there is no true or false answer to the question ‘which sentiment induces action X or state Y for you?’, we asked the annotators to judge as generally as possible (e.g. by prioritising commonly held opinions over their own) but to rely, in the first place, on the context provided by the tweet.

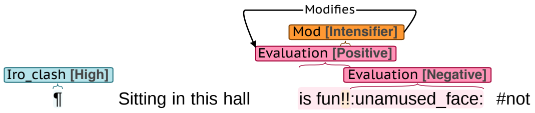
All annotation steps were done using brat, a web-based annotation tool (Stenetorp et al. 2012), some visualisations of which are shown in examples 10 to 12.

(10)



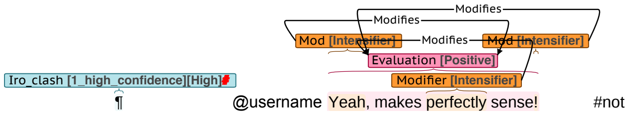
¶ I just love when you test my patience! :white_smiling_face: #Not

(11)



¶ Sitting in this hall is fun!!:unamused_face: #not

(12)

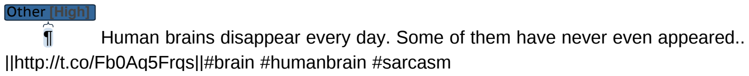


@username Yeah, makes perfectly sense! #not

All three present examples of irony by means of a polarity clash. Example 10 contains a polarity clash between the literal evaluations ‘just love’ and ‘[i]s the best’ and the target ‘you test my patience’, which has been assigned a negative connotation. Like the smiling-face emoticon, the words ‘Just’ and ‘!’ are **modifiers** or elements that intensify the expressed sentiment. In example 11, the polarity opposition takes place between two literal evaluations, namely ‘is fun!!’ and a negative emoticon, hence no implicit sentiment information is required. Sentence 12 is another example of an ironic tweet presenting a polarity contrast, but unlike the two previous examples, the irony cannot be understood from the main text, and no additional context was available to the annotators. In this case, the hashtag *#not* is required, otherwise the evaluation might as well be genuine. Like examples 10 and 11, the sentence also contains modifiers of intensification: ‘yeah’, ‘perfectly’ and ‘!’.

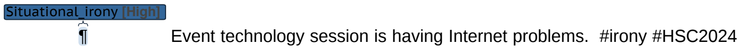
Examples 13 and 14 illustrate other types of irony. While example 14 describes situational irony, example 13 is considered verbal irony, but as opposed to 10, 11 and 12, no polarity contrast is perceived. Finally, 15 is an example of a tweet that is not ironic, the hashtag *#not* does not function as an irony indicator, but is part of the main message.

(13)



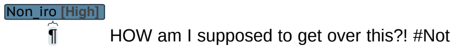
Human brains disappear every day. Some of them have never even appeared.. ||http://t.co/Fb0Aq5Frqs||#brain #humanbrain #sarcasm

(14)



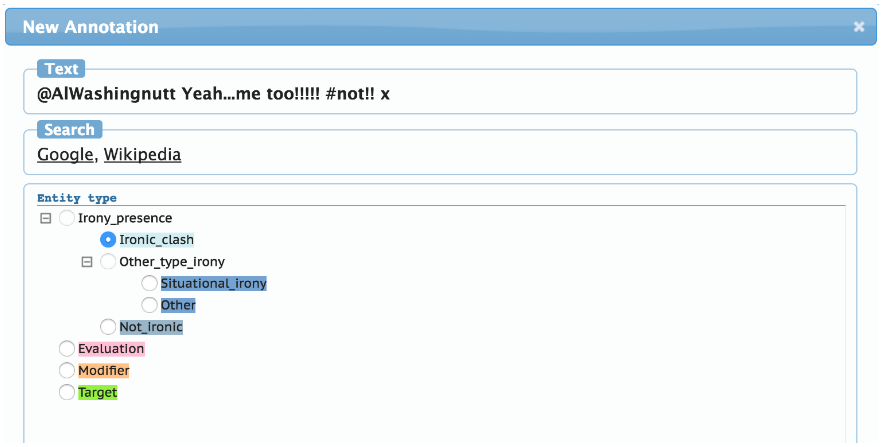
Event technology session is having Internet problems. #irony #HSC2024

(15)



HOW am I supposed to get over this?! #Not

As shown in the examples above, annotators indicated a **confidence score** (i.e. low, medium or high) for each irony annotation to indicate their certainty about the annotation. Whenever *low* or *medium* was indicated for an instance, its annotation received an additional check by one of the experts.



New Annotation ×

Text

@AlWashingnutt Yeah...me too!!!!! #not!! x

Search

Google, Wikipedia

Entity type

⊟ ○ Irony_presence
● Ironic_clash
⊟ ○ Other_type_irony
○ Situational_irony
○ Other
○ Not_ironic
○ Evaluation
○ Modifier
○ Target

Figure 3.3: Screenshot of the annotation scheme in brat.

Figure 3.3 shows a brat visualisation of the main annotation steps as explained in the above paragraphs.

3.2.2 Inter-annotator agreement

The corpus was entirely annotated by three students in linguistics and second-language speakers of English, with each student annotating one third of the entire corpus. All annotations were done using the brat rapid annotation tool (Stenetorp et al. 2012), providing a convenient way to pause and resume the annotation process from whichever machine. To assess the reliability of the annotations, and whether the guidelines allowed to carry out the task consistently, an **inter-annotator agreement study** was set up in two rounds. Firstly, inter-rater agreement was calculated between the authors of the guidelines. The aim of this first study was to test the guidelines for usability and to assess whether changes or additional clarifications were recommended prior to annotation of the entire corpus. For this purpose, a subset of 100 instances from the SemEval-2015 Task *Sentiment Analysis of Figurative Language in Twitter* (Ghosh et al. 2015) dataset were annotated. Based on the results (see Table 3.1), some clarifications and refinements were added to the annotation scheme, including:

- a refinement of the category *other irony* (i.e. subdividing it into *situational irony* and *other verbal irony*);

- a clarification of the concept '(irony) target' to better explain the difference with for instance sentiment targets (e.g. 'I like you');

- a redefinition of the harshness range from a three-point to a two-point scale (i.e. 'harsh' versus 'not harsh'), hence discarding 'slightly harsh'.

After applying the modifications to the annotation scheme, a second agreement study was carried out on a subset of the corpus. It is worth to note that the annotators in this round were three Master's students in linguistics. Each of them annotated the same set of 100 instances to calculate agreement, after which they annotated one third of the entire corpus each.

In both rounds, inter-annotator agreement was calculated at different steps in the annotation process. As metric, we used **Fleiss' kappa** (Fleiss 1971), a widespread statistical measure in the field of computational linguistics for assessing agreement between annotators on categorical ratings (Carletta 1996). Generally, Fleiss' kappa is preferred over Cohen's kappa when assessing the agreement between more than two raters. The measure calculates the degree of agreement in classification over the agreement which would be expected by chance (i.e. when annotators would randomly assign classification labels). In concrete terms, if there is perfect agreement between the annotators, kappa (κ) is one, but if the agreement between annotators is not better than the agreement expected by chance, kappa (κ) is zero.

annotation	**kappa** κ **round 1**	**kappa** κ **round 2**
ironic by clash / other / not ironic	0.55	0.72
hashtag indication	0.60	0.69
harshness	0.32	0.31
polarity contrast (general)	0.62	0.66
polarity contrast (target-evaluation)	0.66	0.55

Table 3.1: Inter-annotator agreement (Kappa scores) obtained in two annotation rounds.

The results of the inter-annotator agreement study are presented in table 3.1. Inter-annotator agreement was calculated for different steps in the annotation process, including the irony type annotation, the indication whether an irony hashtag is required to understand the irony, and the level of harshness in case the tweet is ironic. The last two rows in the table present agreement on the polarity contrast annotation. 'General' refers to the ability of the annotators to indicate two contrasting polarities, regardless of the way the polarity is expressed (i.e. explicit or implicit), while 'target-evaluation' measures the annotators' ability to identify a contrast between explicit and implicit polarities. With the exception of harshness, which proves to be difficult to judge on, kappa scores show a moderate to substantial agreement between the annotators at all annotation steps[3].

Overall, we see that similar or better inter-annotator agreement is obtained after the refinement of the annotation scheme, which has had the largest effect on the irony annotation (i.e. ironic by clash versus other irony versus not ironic). An exception, however, is the annotation of a polarity contrast between targets and evaluations, where the agreement drops from 0.66 to 0.55 between the first and second inter-annotator agreement rounds. An explanation for this drop is that one out of the three annotators in the second round performed less well than the others on this particular annotation. An additional training was therefore given to take away doubts that remained at that point.

Given the difficulty of the task, a kappa score of 0.72 for recognising irony can be interpreted as good reliability. Identifying polarity contrasts between implicit and explicit evaluations, on the other hand, seems to be more difficult, resulting in a moderate kappa of 0.55. A qualitative analysis revealed that identifying polarity contrasts, and especially indicating implicit sentiment (i.e. *targets*) is rather difficult. As mentioned earlier, such judgements are subjective and vary from one person to another, even when public opinion is taken into account. Consequently, cases of doubt were discussed between the annotators or with the experts until a consensus had been found. Once the annotation of the entire corpus was completed, all annotations, including that of implicit sentiment were checked for consistency by one of the experts.

3.2.3 Corpus analysis

In this section, we report the results of a qualitative corpus analysis and present a number of statistics of the annotated data. In total, 3,000 tweets with the hashtags *#irony, #sarcasm* and *#not* were annotated for English based on our fine-grained annotation guidelines (*supra*).

Table 3.2 presents some annotation statistics. As can be inferred from the table, most instances that were labelled as ironic belong to the category *ironic by means of a clash.* When we zoom in on the category *other type of irony*, we see that the subcategory *situational irony* (which encompasses descriptions of ironic situations) constitutes the majority of this annotation class, as compared to other forms of verbal irony.

[3] According to magnitude guidelines by Landis and Koch (1977).

Ironic by means of a clash	**Other type of irony**		**Not ironic**	**Total**
	Situational irony	*Other verbal irony*		
1,728	401	267	604	3,000

Table 3.2: Statistics of the annotated corpus: number of instances per annotation category.

Out of the total of 3,000 tweets, no less than 604 were considered not ironic. This would mean that an irony corpus based on hashtag information as gold labels would contain about 20% of noise. Interestingly, this is twice the number that has been observed by Kunneman et al. (2015). We see three possible explanations for this noise. First, analysis of the data reveals that more than half of the non-ironic tweets contain the hashtag *#not*, which does not always function as an irony indicator since the word has also a grammatical function. In fact, the word 'not' in our corpus is often preceded by a hash-sign, even when used as a negation word (e.g. 'Had no sleep and have got school now #not happy'). Evidently, since *#not* is much less used as a negation word in Dutch, the number of non-ironic tweets mentioning this hashtag is lower. In fact, analysis of a Dutch dataset that is currently under construction shows that 33% of all non-ironic tweets carry the hashtag *#not*, whereas the proportion amounts to 56% in the English corpus. Second, manual analysis showed that irony-related hashtags were sometimes used to refer to the phenomenon (e.g. 'I love that his humor is filled with #irony'). Third, people sometimes added an irony-related hashtag to their tweet which appeared groundless to the annotators, at least given the available context.

Other corpus observations include that given the 1,728 tweets that were annotated as *ironic by means of a polarity contrast*, almost half of them (i.e. 839) required an irony-related hashtag to recognise the irony. In other terms, without such a hashtag, annotators deemed it impossible to recognise the irony (see example 12 in Section 3.2.1). Moreover, more than one third of the instances (i.e. 592 or 34%) were considered harsh, meaning that the text was meant to ridicule someone or something or to cause harm. This is an interesting observation, given that irony literature states that harshness or ridicule, scorn and sharpness could be distinguishing factors between irony and sarcasm, the latter of which is often considered the meaner form of the two (e.g. Attardo 2000, Clift 1999).

In effect, analysis of the harshness annotation revealed that there is a stronger correlation between harshness and the presence of the *#sarcasm* hashtag than compared to the *#irony* and *#not* hashtags. Out of the tweets that were considered harsh, 50% carried the hashtag *#sarcasm*, while this is 38% in the tweets that were not considered harsh. By contrast, the frequency of the other hashtags is more balanced between the harsh and not harsh tweets. Furthermore, tweets that were considered harsh seemed to contain more second-person pronouns (i.e. ‘you’, ‘your’), whereas non-harsh tweets contained more first-person pronouns (i.e. ‘my’, ‘I’). This would suggest that ironic tweets that are harsh and hence directed at a person tend to be signalled by the *#sarcasm* hashtag, whereas other ironic tweets more often refer to the author of the tweet and are more often tagged with the hashtags *#not* and *#irony*. Another observation when looking at harsh tweets in our corpus, is the high frequency of the word ‘thanks’, which was often used to express indignation or disapproval (example 8).

Below tweet level, annotators marked evaluative expressions and *targets*, or implicit sentiment expressions. Confirming our hypothesis, the annotations revealed that most instances of irony showed a polarity contrast between an uttered and implied sentiment, as shown in example 16.

(16) Waking up with a pounding headache is just what I need for this final.

We also observed, however, that sometimes Twitter users make this implied sentiment explicit through a hashtag, an emoji, or words that are syntactically ‘isolated’ from the core text through punctuation. In example 17 for instance, a polarity contrast can be perceived between the explicit ‘I literally love’ and the implicit polarity expression ‘someone throw me in at the deep end’. However, the literal expression ‘#though life’ also reflects the author’s implied sentiment in the first part of the tweet.

(17) I literally love when someone throw me in at the deep end... #though life.

In examples 18 and 19 the implied sentiment is expressed in a separate sentence or an emoticon at the end of the main text.

(18) Yeah that’s always the solution... Doesn’t fix anything!

(19) Picked an excellent day to get my hair done 😫

Although similar realisations of irony are not common in classic examples of irony, they might be typical for Twitter data. In contrast to spoken conversations or genres, including larger text instances (e.g. reviews, blogs), Twitter provides only limited context, as a result of which users, risking to be misunderstood (i.e. the irony is not captured), add extra information to their tweet. Examples like 17 where an implied sentiment is made explicit through the use of a (non-ironic) hashtag may also be considered an idiosyncrasy to express subjectivity (Page 2012).

In sum, the annotations reveal that, when no hashtag is required to recognise the irony in tweets, the polarity contrast is generally realised through an explicit sentiment expression and an implied polarity contained by a *target* (e.g. ‘a pounding headache’). Also, in most cases the explicit sentiment is positive, while the implied one is negative, which is in line with literature stating that irony is used as an indirect or face-protecting strategy to criticise or to mock (e.g. Brown and Levinson 1987, Giora 1995).

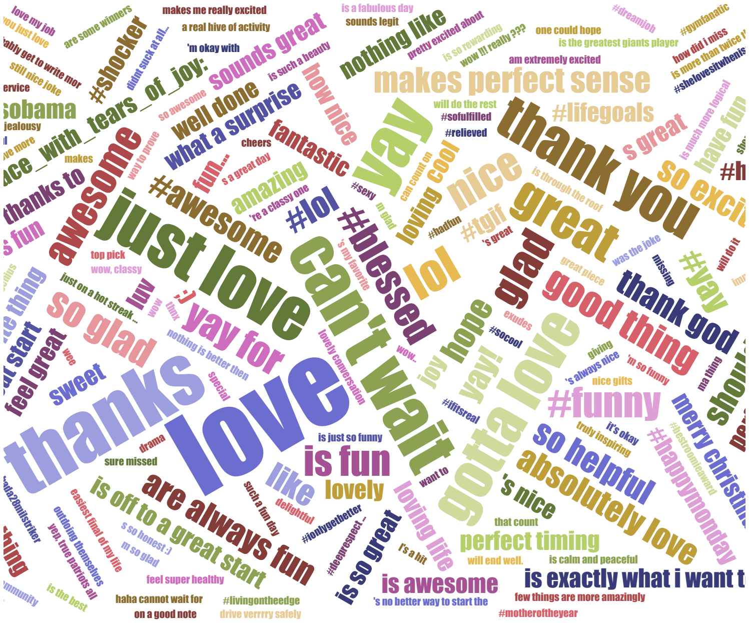


Figure 3.4: Word cloud of explicit positive sentiment expressions in the corpus.

As shown by, among others, Kunneman et al. (2015) and Riloff et al. (2013), ironic utterances are likely to contain ‘sarcasm (or irony) markers’ , including modifiers (e.g. intensifiers, diminishers) to express hyperbole or understatements, interjections (e.g. ‘yeah right’, ‘well’) and highly subjective expressions or words (e.g. ‘I love’, ‘wonderful’). It was also observed in the same studies that positive sentiment expressions to communicate a negative opinion occur much more frequently than inversely. With respect to the latter, analysis of our corpus indeed revealed that, out of a total of 2,090 literal sentiment expressions, 1,614 (77%) were positive and 737 (98%) out of 749 implicit evaluations were negative (see Figure 3.4 for corpus examples of the former category).

This would support the claim that irony is often used as a face-protecting strategy to express negative emotions or venture criticism, as stated by Brown and Levinson (1987) and Giora et al. (2005). Regarding the use of modifiers, we observed that 40% of the ironic instances contained an intensifier, while only 8% contained a diminisher (e.g. '...', 'kinda', 'presumably'). A part-of-speech-based analysis of the corpus revealed that twice the number of interjections were found in the ironic tweets, as compared to the non-ironic ones. The above observations seem to corroborate that modifiers like hyperbole, and interjections are markers of irony in tweets (Hallmann et al. 2016a). However, a more detailed analysis should be conducted to verify whether all observed irony markers apply to our corpus.

3.3 Summary

In this chapter, we introduced our manually annotated irony corpus, which is indispensable for a supervised machine learning approach to irony detection. We collected a set of 3,000 English tweets using the Twitter Search API and irony-related hashtags including *#not*, *#irony* and *#sarcasm*. We established a working definition that is grounded in irony literature and developed a set of coding principles for the fine-grained annotation of irony in social media text. The usability of the scheme was tested and confirmed by conducting a two-staged inter-annotator agreement study. The annotation scheme was applied to the entire corpus so as to create a gold standard dataset for modelling irony.

A qualitative analysis of the annotated corpus revealed that, despite containing an irony hashtag, 20% of the tweets were not ironic, which demonstrates the importance of manual annotations for this task. We found that 72% of the ironic tweets were realised by a polarity contrast, and that the presence of an irony hashtag appeared a necessary clue to discover irony in about half of the tweets. The remaining 30% of ironic tweets consisted of situational irony and other verbal irony. While situational irony contains tweets where an ironic situation is described (see earlier), we observed that *other* verbal irony often contained tweets in which irony is realised by means of a factual opposition or 'false assertion' (Karoui et al. 2015, p. 265).

Although no distinction is drawn between irony and sarcasm in this thesis, we observed that tweets that were annotated as harsh were more frequently tagged with the hashtag *#sarcasm* than with *#irony* and *#not*. They also showed a more frequent use of second person pronouns than tweets that were not considered harsh.

The most interesting observations might, however, concern the text span (or below tweet level) annotations. It was observed that in the category *ironic by means of a clash*, the polarity contrast was in half of the cases realised through an irony-related hashtag causing a negation of the literally expressed sentiment (e.g. ‘Yeah, makes perfectly sense! #not’). However, when no such hashtag was required to recognise the irony, the contrast mostly involved an opposition between an explicit sentiment and a phrase carrying implicit sentiment (i.e. *target*). As the potential of such targets or implicit sentiment expressions for irony detection has also been confirmed in previous work, we will explore how they can be recognised automatically, and to what extent they benefit classification performance (when combined with other information sources). Both research questions will be addressed in Chapter 5 and Chapter 6, respectively.

Prior to investigating how implicit sentiment can be identified automatically, the following chapter presents a series of binary classification experiments to detect irony. For constructing the model, we relied on some of the important insights into the realisation of irony on Twitter that were described in this chapter.

CHAPTER 4

Automatic irony detection

In this chapter, we describe our methodology for automatic irony detection on Twitter. The following sections address the second part of our main research question, namely **'can ironic instances be automatically detected in English tweets and if so, which information sources contribute most to classification performance?'**. To this purpose, we take a supervised machine-learning approach and investigate the informativeness of a varied feature set.

We start this chapter by presenting the experimental corpus, after which we elaborate on the development of our irony detection pipeline and explore the benefits of combining different feature groups for the task.

4.1 Introduction

In the past few years, research in natural language processing (NLP) has seen various attempts to tackle automatic irony detection (see Chapter 2). As described in the survey by Joshi, Bhattacharyya and Carman (2016), recent approaches to irony can be roughly classified into rule-based, and (supervised and unsupervised) machine-learning-based. While rule-based approaches often rely on lexicons and word-based information, popular information sources in machine learning are bags of words, syntactic patterns, polarity information and semantic information provided by word embeddings. Twitter has been a popular data genre for this task, as it is easily accessible and contains self-describing hashtags or meta-communicative irony clues (e.g. *#irony*), which facilitate data collection.

In this research, we explore automatic irony detection based on **supervised learning**, meaning that a machine learning algorithm is trained with a set of manually labelled instances. The algorithm analyses these training data and produces a function that allows to compare an unseen instance against the training data and predict a relevant class label for this new instance. Depending on the task, such class labels are numeric values (e.g. a sentiment score between -5 and 5), or categorical labels from a predefined set (e.g. *positive*, *negative*, *neutral*). In binary classification tasks including the present, there are only two such categorical labels (generally presented by 0 and 1), indicating for each instance whether it belongs to a certain category (1) or not (0).

For the experiments, we make use of a support vector machine (SVM) as implemented in the LIBSVM library (Chang and Lin 2011). We chose an SVM as the classification algorithm, since support vector machines have proven their suitability for similar tasks and have been successfully implemented with large feature sets (Joshi, Bhattacharyya and Carman 2016).

4.2 Experimental corpus

As we described in Chapter 3, our corpus comprises 3,000 tweets that were manually annotated. About 20% of them were considered non-ironic and were added to the negative class for our binary classification experiments, which leaves 2,396 ironic and 604 non-ironic tweets. To balance the class distribution in our experimental corpus, we expanded the latter with a set of non-ironic tweets (1,792 to be precise) from a background corpus. As such, the experimental corpus contains 4,792 English tweets and shows a balanced class distribution (i.e. ironic versus not ironic). Next, the corpus was randomly split into a training and test set of respectively 80% (3,834 tweets) and 20% (958 tweets), each showing a balanced class distribution. While the former was used for feature engineering and classifier optimisation purposes, the latter functioned as a held-out test set to evaluate and report classification performance.

In Chapter 3, we elaborated on the construction and fine-grained annotation of our irony corpus. It is important to remind the reader that the data were collected using the hashtags *#irony, #sarcasm* and *#not* (we refer to this corpus as 'the hashtag corpus' throughout this chapter). Manual annotations revealed that 80% of the tweets were actually ironic (i.e. 58% ironic by clash, 13% situational irony, 9% other irony), whereas 20% were not (see Chapter 3 for an explanation of this percentage). To balance the class distribution in our experimental corpus, a set of non-ironic tweets were added from a background corpus. The tweets in this corpus were collected from the same set of Twitter users as in the hashtag corpus, and within the same time span. It is important to note that these tweets do not contain irony-related hashtags (as opposed to the non-ironic tweets in the hashtag corpus), and were manually filtered from ironic tweets.

	ironic by clash	**other type of irony**		**not ironic**	**not ironic**
		situational irony	*other verbal irony*	*(hashtag corpus)*	*(backgr. corpus)*
	1,728	401	267	604	1,792
total	**2,396**			**2,396**	

Table 4.1: Experimental corpus statistics: number of instances per annotation category plus non-ironic tweets from a background corpus.

Table 4.1 presents the experimental corpus comprising different irony categories as annotated in the hashtag corpus (see Chapter 3), and 1,792 non-ironic tweets from a background corpus that were included to obtain a balanced class distribution. This allows us to compare the experimental results with related work on automatic irony detection, which mostly work with balanced irony datasets.

4.3 Preprocessing and feature engineering

Data preprocessing is an important step in machine learning. It includes the steps that are required to extract text characteristics that might benefit a classifier. Standard preprocessing steps in natural language processing include tokenisation (i.e. segmentation of a text into words), part-of-speech tagging (i.e. assigning grammatical categories to the tokens), lemmatisation (i.e. converting words to their canonical form), dependency parsing (i.e. syntactic decomposition), and named entity recognition (i.e. location and classification of persons, locations, organisations, etc. in text).

Prior to feature extraction, a number of preprocessing steps were taken. Preprocessing refers to all steps that are needed for formatting and cleaning the collected tweets and enriching the data with the linguistic information required for feature engineering, including the following:

- **Tokenisation:** lexical analysis that divides a text into a sequence of meaningful elements (or *tokens*), which roughly correspond to words.

- **Part-of-Speech tagging:** lexical analysis that assigns part-of-speech categories to each token, such as N (noun), and A (adjective), but also Twitter-specific categories such as # (hashtag) and E (emoticon).
- **Lemmatisation:** determining the lemma or dictionary form of a word based on its intended meaning (i.e. part-of-speech tag)
- **Named entity recognition (NER):** linguistic process that labels sequences of words that are the names of entities including people, companies, events, countries, etc.

Tokenisation and PoS-tagging were done using the Carnegie Mellon University Twitter NLP Tool (Gimpel et al. 2011), which was trained on user-generated content. For lack of a reliable Twitter-specific lemmatiser, we made use of the LeTs Preprocess (Van de Kauter et al. 2013). Finally, named entity recognition was performed using the Twitter named entity recogniser by Ritter et al. (2011).

Additionally, all tweets were cleaned (e.g. replacement of HTML-escaped characters and multiple white spaces) and a number of (shallow) normalisation steps were introduced to decrease feature sparseness. In concrete terms, all hyperlinks and @-replies in the tweets were normalised to 'http://someurl' and '@someuser', respectively, and abbreviations were replaced by their full form based on an English abbreviation dictionary[1] (e.g. 'w/e' → 'whatever'). Furthermore, variations in suspension dots were normalised to thee dots (e.g. '.....' → '. . . '), multiple white spaces were reduced to a single space, and vertical bars or *pipes* were discarded. Finally, we removed irony-related hashtags that were used to collect the data (i.e. *#irony, #sarcasm, #not*).

Following preprocessing, another crucial step in machine learning is feature engineering. Machine learning algorithms represent each instance (i.e. a text in the training corpus or a text for which a prediction has to be made) by means of a set of *features*. Features are pieces of information that (are expected to) suit a particular task. They can be numeric (e.g. the number of tokens in an instance), or categorical (e.g. birth place of the author). The latter are called binary features if they have only two possible values, mostly **1** (i.e. the feature is present) or **0** (i.e. the feature is absent). An example of a binary feature would be whether or not an exclamation mark is present in the instance under investigation.

To train our irony detection system, all tweets were represented by a number of features that potentially capture ironic text. Based on the information

[1] http://www.chatslang.com/terms/abbreviations.

they provide, they can be divided into four groups, namely **lexical** features, **syntactic** features, **sentiment** lexicon features, and **semantic** features. The feature groups bring together a varied set of information sources, some of which have proven their relevance for this type of tasks, including bags of words (e.g. Liebrecht et al. 2013, Reyes et al. 2013), part-of-speech information (e.g. Reyes and Rosso 2012), punctuation and word-shape features (e.g. Tsur et al. 2010), interjections and polarity imbalance (e.g. Buschmeier et al. 2014), sentiment lexicon features (e.g. Bouazizi and Ohtsuki 2016, Riloff et al. 2013, Van Hee et al. 2016b), and semantic similarity based on Word2Vec embeddings (Joshi, Tripathi, Patel, Bhattacharyya and Carman 2016). The following paragraphs present a detailed overview of all features contained by the different feature groups we defined.

4.3.1 Lexical features

A first set of lexical features are **bags of words** (*bow*). Bow features present a tweet as a 'bag' of lexical units or sequences (called *n-grams*), formed by words or characters. The usefulness of n-gram features for similar tasks has been demonstrated in the past (e.g. Jasso López and Meza Ruiz 2016). We included token-, as well as character-based n-grams, since user-generated content is often noisy (i.e. containing grammatical errors, creative spelling, etc.) and the latter provide some abstraction from the word level. For instance, character-based n-grams capture the common sequence 'ever' in different (i.e. erronic) spellings of the word 'forever' (e.g. '4-ever', 'foreverrr', 'forever'). By contrast, word n-grams would consider them as three different tokens.

Based on preliminary experiments on our dataset, the following n-grams were extracted as binary, sparse (i.e. only features with feature value 1 are included in the feature vector) features:

- word unigrams and bigrams (*w1g, w2g*)
- character trigrams and fourgrams, including token boundaries (*ch3g, ch4g*)

The n-grams were created using raw words rather than lemmas or canonical forms to retain morphological information, and punctuation marks and emoticons were included as well. N-grams that occurred only once in the training corpus were discarded to reduce sparsity, resulting in a total of 8,680 token and 27,171 character n-gram features.

Secondly, the lexical features contain a set of shape or **word form features**, some of which are binary (*), while others are numeric (-). The binary features have either 1 or 0 as feature value, whereas numeric features represent normalised floats which are divided by the tweet length in tokens (except for the *tweet length* feature). The following word shape features were exploited:

* character flooding (i.e. 2 or more repetitions of the same character)

* punctuation flooding (i.e. 1 or more repetition of the same punctuation mark)

* punctuation last token

- number of punctuation marks
- number of capitalised words
- number of hashtag words
- number of interjections
- hashtag-to-word ratio
- emoticon frequency
- tweet length

A third set of lexical features include **conditional n-gram probabilities** based on language models that were constructed from two (i.e. ironic and non-ironic) background corpora. Although often exploited in machine translation research (e.g. Bojar et al. 2016), language model information as features is, to our knowledge, novel in irony detection. The language models were created with KENLM (Heafield et al. 2013) and are trained on an ironic and a non-ironic background corpus. Data for both corpora were collected using the Twitter Search API[2]. To retrieve ironic tweets, we searched with the hashtags *#irony, #sarcasm* and *#not*. Non-ironic tweets were collected without any specific search query so as to obtain a set that is as general as possible. Tweets containing irony-related hashtags were removed. The corpora were cleaned and preprocessed in the same way as the experimental corpus, which resulted in 204,237 ironic and 921,891 non-ironic tweets. Prior to constructing language models, all tweets were split into sentences, and two equally sized corpora (i.e. 354,565 ironic and 354,565 non-ironic sentences) were created. As features we extracted two log probabilities between source and target sentences, indicating how probable a tweet (i.e. all sentences in that tweet) is to appear in either an ironic or non-ironic corpus. Two additional features include the number of out-of-vocabulary (OOV)

[2] The data were collected between April 2016 and January 2017 by crawling Twitter at regular intervals.

words a tweet contains based on each language model. OOV-words are words that appear in the tweet, but that were not seen yet in the training corpus of the language model.

4.3.2 Syntactic features

Two numerical and one binary feature were included to incorporate syntactic information. Part-of-speech tagging and named entity recognition were performed as preprocessing steps.

- Part-of-speech features: four features for each of the 25 tags used by the Twitter part-of-speech tagger by Gimpel et al. (2011). These indicate for each pos-tag (i) whether it occurs in the tweet or not, (ii) whether the tag occurs 0, 1, or ≥ 2 times, (iii) the frequency of the tag in absolute numbers and as a percentage (iv).

* Temporal clash: a binary feature indicating a clash or contrast between verb tenses in the tweet (following the example of Reyes et al. 2013). We used the LeTs Preprocess part-of-speech tagger (Van de Kauter et al. 2013) since it provides verb tense information, as opposed to the Twitter tagger by Gimpel et al. (2011).

- Named entity features: four features indicating the presence of named entities in a tweet: one binary feature indicating the presence of a named entity in the tweet, and three numeric features, indicating (i) the number of named entities in the text, (ii) the number and (iii) frequency of tokens that are part of a named entity.

4.3.3 Sentiment lexicon features

Six sentiment-lexicon-based features were included to investigate whether sentiment clues provide valuable information for automatic irony detection. For this purpose, we made use of existing sentiment lexicons for English: AFINN (Nielsen 2011), General Inquirer (Stone et al. 1966), the MPQA subjectivity lexicon (Wilson et al. 2005), the NRC emotion lexicon (Mohammad and Turney 2013), and Bing Liu's opinion lexicon (Liu et al. 2005). All of the above lexicons are commonly used in sentiment analysis research (Cambria et al. 2017), and their validity was confirmed in earlier experiments (Van Hee et al. 2014) where a preliminary study revealed that, by using merely the above lexicons as information sources, about 60% of the training data could be assigned the correct sentiment label.

In addition to these well-known sentiment resources, we included Hogenboom's emoticon lexicon (Hogenboom et al. 2015), and Kralj Novak's emoji lexicon (Kralj Novak et al. 2015), both tailored to social media data.

For each of the above lexicons, five numeric and one binary feature were derived, including:

- the number of positive, negative and neutral lexicon words averaged over text length;
- the overall tweet polarity (i.e. the sum of the values of the identified sentiment words);
- the difference between the highest positive and lowest negative sentiment values;

* a binary feature indicating the presence of a polarity contrast between two lexicon words (i.e. at least one positive and one negative lexicon word are present).

With the latter feature, we aim to capture an explicit polarity contrast in a tweet, as the annotations (see Chapter 3) revealed that about 70% of ironic tweets show a clash between two (explicit or implicit) polarities in the text. It is important to note, however, that the feature will not be able to signal implicit polarity contrasts, which would require world knowledge in addition to sentiment lexicon information.

The sentiment lexicon features were extracted in two ways: (i) by considering all tokens in the instance and (ii) by taking into account only hashtag tokens, without the hashtag (e.g. *lovely* from *#lovely*). We took negation clues into account by flipping the polarity of a sentiment word when occurring within a window of one word to the right of a negation word (*not, never, don't*, etc.)

4.3.4 Semantic features

The last feature group includes **semantic features**. Our hypothesis is that ironic tweets might differ semantically from their non-ironic counterparts (e.g. some topics or themes are more prone to irony use than others). To verify this assumption, we utilised semantic word clusters created from a large background corpus. The clusters were defined based on word embeddings generated with Word2Vec (Mikolov et al. 2013) and were implemented as one binary feature per cluster, indicating whether a word contained in that cluster occurred in the tweet. The following are two examples of such clusters:

(20) `college, degree, classes, dissertation, essay, headache, insomnia, midterm, migraine, monday, motivation, mood, papers, revision, presentation`

(21) `headaches, health, hormones, ibuprofen, illness, infected, lung, medication, organs, pain, rash, recovery, suffer, therapy, trauma`

The word embeddings were generated from an English background corpus comprising 1,126,128 (ironic + non-ironic) tweets (see details on the language model features). We ran the Word2Vec algorithm on this corpus, applying the continuous bag-of-words model, a context size of 5, a word vector dimensionality of 100 features, and a cluster size (k) of 200. For each parameter of the algorithm, different values were tested and evaluated by means of 10-fold cross validation experiments on the training data.

4.3.5 Feature statistics

In summary, four feature groups were defined for the experiments. All groups and the number of individual features they contain are presented in Table 4.2. The combined feature vectors consist of 36,270 individual features, the majority of which (> 98%) are part of the binary bag-of-words features within the lexical feature group. In Section 4.5, we evaluate the predictive power of the individual feature groups and test a series of feature group combinations by means of binary classification experiments.

	feature group			
	lexical	**sentiment**	**semantic**	**syntactic**
# features	35,869	96	200	105

Table 4.2: Number of features per feature group.

Before constructing models, we first scaled our feature vectors (which is considered good practice when working with SVM (Hsu et al. 2003)), which means that all features were linearly mapped to the range [0,1]. As stated by Hsu et al. (2003), two important advantages of scaling before applying SVM include i) avoiding feature values in greater numeric ranges dominating those in smaller ranges, and ii) reducing numerical complexity during the construction of the model.

4.4 Experimental setup

The following paragraphs describe a set of binary classification experiments for automatic irony detection in tweets. Our training corpus consists of 3,834 English Tweets, while the held-out test corpus contains 958 tweets (*supra*). Feature extraction was done after removal of irony-related hashtags (e.g. *#sarcasm, #irony, #not*) in both corpora.

The feature types (i.e. *feature groups*) were tested individually and in combination to see whether they provide the classifier with complementary information. The results presented in this section are obtained through cross-validation on the training data to obtain suitable parameter settings for the classifier, and by evaluating the resulting model on a held-out test set comprising 958 tweets.

4.4.1 Machine learning method

For our experiments, we made use of a support vector machine (SVM), a machine learning classifier for binary classification tasks. It can also be used for multiclass classification problems when applied as a one-against-the-rest classifier per class, or a binary classifier for every pair of classes. We opted for an SVM because of its acknowledged robust generalisation ability (Cortes and Vapnik 1995) and because its performance for similar tasks has been recognised (e.g. Joshi, Bhattacharyya and Carman 2016, Reyes et al. 2013, Riloff et al. 2013).

In essence, a SVM classifies data by finding the linear decision boundary (or *hyperplane*) that separates the data points of two classes in a training set. To this end, all training instances are represented as a vector in the feature space. Such a vector is a sequence of *features* or textual characteristics of an instance that are indicative of the class label (e.g. words like 'sure' and 'soo' in the current task). New (i.e. unseen) instances are classified by mapping them to this feature space and assigning a label depending on their position with respect to the hyperplane (e.g. if it is below the hyperplane, it receives the class label 0, if it is above, the label is 1). The implementation used in our experiments is LIBSVM, a popular SVM-library by Chang and Lin (2011) that supports various formulations for tasks including classification and regression.

We performed binary SVM classification using the default *radial basis function* (i.e. RBF or *Gaussian*) kernel, the performance of which equals that of a linear kernel if it is properly tuned (Keerthi and Lin 2003). Preliminary experiments on our dataset showed even better results using RBF. We optimised the SVM's cost value C and RBF's single parameter γ. Defining an appropriate cost value is essential for building the model, as it trades off misclassification of the training examples against complexity of the decision hyperplane. This means that, when C is high, the decision boundary is supposed to fit most if not all training instances, whereas low C aims for a maximum separating distance between the two classes, and allows for some misclassifications on the training data to serve more generalisation as the higher purpose. The gamma parameter γ defines the impact or influence of a single training example on the decision boundary. Small gamma implies a kernel with a large variance, so the decision boundary will depend on many training instances (which might cause overfitting).

Given the importance of parameter optimisation to obtain good SVM models (Chang and Lin 2011), optimal C and γ values were defined for each experiment exploiting a different feature group or feature group combination. For this purpose, a cross-validated grid search was performed across the complete training data. During the parametrisation, γ was varied between 2^{-15} and 2^3 (stepping by factor 4), while C was varied between 2^{-5} and 2^{15} (stepping by factor 4). The optimal parameter settings were used to build a model for each feature setup using all the training data, which was evaluated on the held-out test set.

4.4.2 Classifier evaluation

For each experiment, we report the classifier's cross-validated performance on the training data and its performance on the held-out test set. As evaluation metrics, we report **accuracy, precision, recall** and **F_1-score**, indicating how well the classifier detects irony. Accuracy is the proportion of correctly classified instances considering all labels, i.e. the correct positive and negative predictions (equation 4.1).

$$accuracy = \frac{true\ positives\ +\ true\ negatives}{total\ number\ of\ instances} \tag{4.1}$$

Although being a popular evaluation metric, accuracy might be misleading if the dataset is imbalanced or skewed (which is known as the 'accuracy paradox' (Fernandes et al. 2010)) as it weighs class labels proportionally to their frequency. For instance, when applied to a test set comprising 80% negative and 20% positive data, a classifier that predicts all instances as negative would achieve an accuracy of 80%, while it actually performs poorly on the positive class. Whether or not the metric is preferred, however, depends on the dataset and task at hand. In case it is preferable to report scores per class label (e.g. for the positive label in a binary classification tasks), precision, recall and F$_1$-measure can be used as evaluation metrics. While precision indicates how accurately a system works, recall gives an idea of its sensitivity. More specifically, precision is calculated by dividing the number of correct predictions (i.e. *true positives*) by the total number of instances that were predicted to be positive by the classifier, including those that were wrongly considered positive (i.e. *false positives*) (equation 4.2). Recall is the ratio of correct predictions (i.e. *true positives*) to the total of positive instances in the test set, including the ones that were ignored by the classifier (i.e. *false negatives*) (equation 4.3).

$$precision = \frac{true\ positives}{true\ positives\ +\ false\ positives} \tag{4.2}$$

$$recall = \frac{true\ positives}{true\ positives\ +\ false\ negatives} \tag{4.3}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision\ +\ recall} \tag{4.4}$$

Finally, F$_1$-score is the harmonic mean of precision and recall (equation 4.4). In multi-class or multi-label classification tasks, it is an average of the F$_1$-scores per class (see Chapter 7). In binary classification or detection tasks like the present, F$_1$-score is calculated on the positive (i.e. ironic) instances only.

4.5 Baselines and experimental results

In this section, we describe our experiments to explore the feasibility of automatic irony detection in English tweets. The experimental setup can be broken down into three parts, starting with the implementation of three baselines to evaluate the performance of our models, followed by the development of a binary classifier for i) each individual feature group, and ii) a set of feature group combinations (see Section 4.5.2). A summary of our findings is presented in Section 4.5.3.

4.5.1 Baseline classifiers

Three straightforward baselines were implemented against which the performance of our irony detection model can be compared: a random class baseline and two n-gram baselines. The random class baseline is a classifier which randomly assigns a class label (i.e. ironic or not ironic) to each instance. Next, we calculated the performance of two classifiers based on token unigram (*w1g*) and bigram (*w2g*) features (i.e. single words and combinations of two words, respectively), and character trigram (*ch3g*) and fourgran (*ch4g*) features (i.e. combinations of three and four characters). Despite their simplicity and universal character, n-gram features have proven to work well for this task (e.g. Liebrecht et al. 2013, Reyes et al. 2013). Hyperparameter optimisation is crucial to the good functioning of the algorithm, hence it was also applied in the baseline experiments, except for *random class.*

baseline	**optimised parameters**	**cross-validated accuracy**
random class	*n.a.*	n.a.
w1g + w2g	C=2^1, γ=2^{-5}	65.60%
ch3g + ch4g	C=2^1, γ=2^{-7}	66.25%

Table 4.3: Cross-validated accuracy and optimised parameters of the baselines.

baseline	**accuracy**	**precision**	**recall**	$\mathbf{F_1}$
random class	50.52%	51.14%	50.72%	50.93%
w1g + w2g	66.60%	67.30%	66.19%	66.74%
ch3g + ch4g	**68.37%**	**69.20%**	**67.63%**	**68.40%**

Table 4.4: Classification results of the baselines (obtained on the test set).

Tables 4.3 and 4.4 display the baseline scores by cross-validation on the training and testing on the held-out test set, respectively. As mentioned earlier, three baselines were implemented, including random class, word n-grams (unigrams + bigrams), and character n-grams. While the random class baseline clearly benefits from the balanced class distribution in the test set, we find that the n-gram classifiers already present strong baselines that show a good balance between precision and recall.

4.5.2 Experimental results

Varied feature types (or groups) for automatic irony detection were exploited and evaluated for this section: evaluation was done on the groups in isolation, as well as in different combined setups.

Individual feature groups

We first tested the importance of the individual feature groups. For this purpose, four models were built on the basis of lexical, syntactic, sentiment and semantic features. Table 4.5 presents cross-validated results obtained for each feature group, together with the optimised C and γ-values for that setup, while Table 4.6 displays the scores of the individual feature groups on the held-out test set. To facilitate comparison, the baseline scores are included in grey. The best results are indicated in bold.

feature group	**optimised parameters**	**cross-validated accuracy**
lexical	C=2^{3}, γ=2^{-11}	**66.69**
sentiment	C=2^{11}, γ=2^{-5}	61.01%
semantic	C=2^{1}, γ=2^{-5}	63.56%
syntactic	C=2^{15}, γ=2^{-13}	63.62%

Table 4.5: Cross-validated accuracy and optimised parameters of the individual feature groups.

feature group	**accuracy**	**precision**	**recall**	**F$_1$**
lexical	**66.81%**	**67.43%**	66.60%	**67.01%**
sentiment	58.77%	61.54%	49.48%	54.86%
semantic	63.05%	63.67%	62.89%	63.28%
syntactic	64.82%	64.18%	**69.07%**	66.53 %
baselines				
random class	50.52%	51.14%	50.72%	50.93%
w1g + w2g	66.60%	67.30%	66.19%	66.74%
ch3g + ch4g	68.37%	69.20%	67.63%	68.40%

Table 4.6: Experimental results of the individual feature groups (obtained on the test set).

Table 4.6 confirms the strong baseline that present n-gram features, given that none of the feature groups outperforms the character n-gram baseline. Character n-gram features outperforming the lexical feature group (which contains a fair number of other lexical clues in addition to character n-grams) might suggest that the former work better for irony detection. This seems counter-intuitive, since the lexical feature group includes information which has proven its usefulness for irony detection in related research (e.g. punctuation, flooding). An explanation would be that the strength of a number of individual features in the lexical feature group (potentially the most informative ones) is undermined by the abundance of features in the group.

The lexical features group does, however, outperform the token n-gram baseline. While this would suggest that lexical features are more informative for irony detection than the other feature groups, it is noteworthy that all other feature groups (i.e. syntactic, sentiment and semantic) contain much less features than the lexical group, and that the features are not directly derived from the training data, as opposed to the bag-of-words features in the lexical group.

Recall being less than 50% for the sentiment lexicon-based features shows that, when using merely explicit sentiment clues, about half of the ironic tweets are missed by the classifier. This confirms our hypothesis that explicit sentiment information is insufficient to distinguish between ironic and non-ironic text, and this observation is in line with the findings of Riloff et al. (2013), reporting F_1-scores between 14% and 47% when using merely sentiment lexicons for irony detection. Interestingly, Barbieri and Saggion (2014) and Farías et al. (2016) observed that sentiment features do perform well in irony detection. However, both distinguished ironic tweets from particular genres in non-ironic tweets, based on specific topics, namely *humor, education, newspaper* and *politics*. Given that the topics, especially the latter three, are less likely to contain highly subjective words, sentiment lexicon feature might indeed be helpful to distinguish them from ironic text, which is typically strongly subjective (Grice 1978). By contrast, the non-ironic tweets in our dataset were randomly collected and are therefore as likely to be subjective (see examples 22 and 23) as ironic tweets.

(22) Why didn't I start watching the tudors earlier? #iloveit

(23) Sad that when someone drinks they treat you like shit and won't talk to you.

A more general observation is that classification performance on the test set does not differ much from that on the training data, showing sometimes even slightly less error than the latter. This could suggest that, albeit being randomly split, the test set might better fit the model than the training set.

Based on the raw results, we can conclude that overall, lexical features perform best for the task (F_1= 67%). However, the best recall (69%) is obtained using syntactic features, a score that even outperforms the naïve, yet strong baselines. A qualitative analysis of the classifiers' output indeed revealed that lexical features are not the holy grail to irony detection, and that each feature group has its own strength, by identifying a particular type or realisation of irony. We observed for instance that lexical features are strong predictors of irony (especially *ironic by clash*) in short tweets and tweets containing clues of exaggeration (e.g. character repetition, see example 24), while sentiment features often capture *ironic by clash* instances that are very subjective or expressive (example 25). As opposed to lexical features, syntactic features seem better at predicting irony in (rather) long tweets and tweets containing *other verbal irony* (example 26). Finally, semantic features contribute most to the detection of situational irony (example 27).

(24) Loooovvveeee when my phone gets wiped -.-

(25) Me and my dad watch that bangla channel for bants.. loool we try to figure out what theyr saying.. this is the life.

(26) Cards and Panthers? or watch my own team play a better sport...... hmmm touch choice LOL

(27) SO there used to be a crossfire place here ...#pizzawins

In sum, the experiments show that although only lexical features outperform the word n-gram baseline, semantic, syntactic and (to a lesser extent) sentiment features show to be good indicators for irony as well. This is why, in the next section, we investigate the potential of combining the aforementioned feature groups for this task. The individual feature groups we exploited in this section may provide us with insights into the type of information that is important for irony detection. It would be interesting, however, to extend this work by performing individual feature selection and investigate the informativeness of individual features in each feature group (e.g. particular part-of-speech tags, n-grams or semantic clusters). This could also provide better insights into the performance of the sentiment lexicon features (e.g. which lexicons provide the best features?), which scored least well in these experiments. This is beyond the scope of the current thesis, however, and will constitute an important direction for future research.

Feature group combinations

While performance of the individual feature groups for irony detection was evaluated in the previous section, the following paragraphs shed light on the added value of combined feature groups for this task.

Tables 4.7 and 4.8 present the results of a binary irony classifier exploiting a combination of feature groups. While the former displays the optimal parameters and accuracies obtained through cross validation on the training set, the latter presents the results obtained on the held-out test set. The best individual feature group (i.e. lexical) and the character n-gram baselines were also included in the table for the purpose of comparison.

feature group combination	**optimised parameters**	**cross-validated accuracy**
lex + sent	$C=2^{1}$, $\gamma=2^{-7}$	67.03%
lex + sem	$C=2^{1}$, $\gamma=2^{-7}$	67.45%
lex + synt	$C=2^{3}$, $\gamma=2^{-7}$	67.48%
sent + sem	$C=2^{1}$, $\gamma=2^{-5}$	64.84%
sent + synt	$C=2^{5}$, $\gamma=2^{-7}$	64.96%
sem + synt	$C=2^{1}$, $\gamma=2^{-5}$	66.17%
lex + sent + sem	$C=2^{3}$, $\gamma=2^{-7}$	67.63%
lex + sent + synt	$C=2^{1}$, $\gamma=2^{-7}$	67.74%
lex + sem + synt	$C=2^{1}$, $\gamma=2^{-7}$	67.81%
sent + sem + synt	$C=2^{1}$, $\gamma=2^{-5}$	66.82%
lex + sent + sem + synt	$C=2^{1}$, $\gamma=2^{-7}$	**68.00%**

Table 4.7: Cross-validated accuracy and optimised parameters of the combined feature groups: lexical (*lex*), sentiment (*sent*), semantic (*sem*), and syntactic (*synt*).

feature group combination	**accuracy**	**precision**	**recall**	**F_1**
lex + sent	69.21%	**69.79%**	69.07%	67.43%
lex + sem	69.21%	69.31%	70.31%	69.81%
lex + synt	**69.42%**	69.43%	70.72%	70.07%
sent + sem	66.08%	67.94%	62.47%	65.09%
sent + synt	64.72%	64.97%	65.77%	65.37%
sem + synt	66.70%	67.22%	66.80%	67.01%
lex + sent + sem	69.52%	69.52%	69.52%	69.52%
lex + sent + synt	69.10%	69.33%	69.90%	69.61%
lex + sem + synt	69.21%	68.92%	**71.34%**	**70.11%**
sent + sem + synt	66.39%	67.45%	64.95%	66.18%
lex + sent + sem + synt	69.00%	68.95%	70.52%	69.72%
baselines				
lexical	66.81%	67.43%	66.60%	67.01%
ch3g + ch4g	68.37%	69.20%	67.63%	68.40%

Table 4.8: Experimental results of the combined feature groups.

From the results in the table, we can deduce that combining feature types improves classification performance, given that more than half of the combinations present an improvement over the character *n*-gram baseline and lexical features alone. In particular, combining lexical with semantic and syntactic features seems to work well for irony detection, yielding a top F_1-score of 70.11%, although a similar score is achieved when combining lexical with syntactic features (i.e. $F_1 = 70.07\%$). Like the individual feature group experiments (see Table 4.6), the error rate on the test set compares to the error rate on the training data.

Judging from the raw performance results, it proves beneficial to combine lexical with semantic and syntactic features for the classifier when compared to character n-grams. By making use of the bootstrap resampling test (Noreen 1989), we investigated whether the results of the two systems show a significant difference. To this purpose, bootstrap samples ($n = 958$) with replacement were randomly drawn from the output of both systems (i.e. the character n-gram baseline and the best combined system). This was done 10,000 times and F_1-score was calculated for each sample.

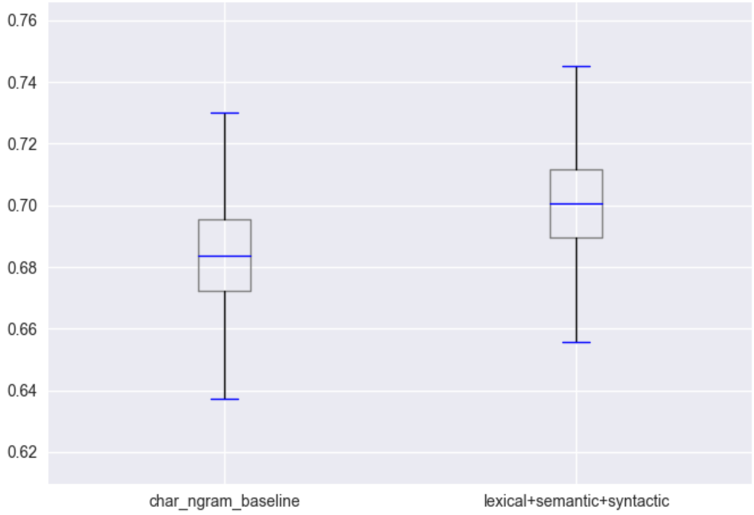


Figure 4.1: Box plots displaying variation in the bootstrap samples for both systems (Y-axis: F_1-score on the positive class).

We subsequently applied a paired samples t-test to compare the mean scores and standard error over all sample scores (see Figure 4.1) and observed a significant difference ($p < 0.05$) between the two systems.

Combining classifier outputs

Table 4.8 shows that generally, combined feature groups only slightly outperform lexical features alone, although they provide totally different types of information. We wanted to verify whether combining the output of two systems proves more beneficial to the classification performance than combining features into one model.

To this purpose, we evaluated classification performance when the classifier would be informed by the output (i.e. predicted class labels) of two different models. Not considering the character baseline, the lexical system performs best, hence we considered its output the starting point and investigated whether other systems would provide supplementary information, that is, by finding ironic tweets that the lexical system overlooks. For each instance, we looked at the predictions made by the lexical system and informed it with the prediction for that instance by one of the other systems (i.e. sentiment, semantic or syntactic). More precisely, whenever the lexical system predicted an instance as non-ironic (i.e. a negative class label was assigned), the instance was processed by one of the other systems to see whether a positive label was assigned by it. If so, the final label for the instance was positive (i.e. *ironic*).

Finally, we combined the output of all classifiers; only when all systems predicted an instance as ironic, it was classified as such.

setup	**accuracy**	**precision**	**recall**	**F$_1$**
lex & sent	62.94%	60.32%	78.35%	68.16%
lex & sem	**67.12%**	63.58%	**82.06%**	**71.65%**
lex & synt	66.08%	62.82%	80.82%	70.69%
lex & sent, sem, synt	59.08%	**78.18%**	26.60%	39.69%

Table 4.9: Results obtained when combining the output of the lexical system with that of the other feature groups.

As can be deduced from Table 4.9, informing the lexical system with the output of the semantic features classifier yields an improvement of 1.5 points over the best combined feature groups system (F$_1$= 71.65% *versus* 70.11%). This could indicate that different types of features are likely to provide complementary information for irony detection, on the condition, however, that small feature groups are not overshadowed by large ones (e.g. bag-of-words features). Taking into account the predictions of all systems results, logically, in a very high precision, but at the expense of recall.

4.5.3 Analysis

To sum up, for the experiments described in this section, the positive class encompasses different types of irony as distinguished in our annotation scheme, namely *ironic by clash, situational irony*, and *other verbal irony.* Also important to note is that the non-ironic data comprises tweets from the hashtag corpus (i.e. whose instances originally carried an irony-related hashtag), and non-ironic tweets from a background corpus (i.e. devoid of such hashtags).

In the annotations section (see Chapter 3), we found that most ironic tweets in our corpus (i.e. 72%) show a contrast between a positive and a negative polarity. In the following paragraphs, we aim to verify whether this category is also the most likely to be recognised automatically, as compared to other types of irony. Another observation that results from the annotations is that 11% of all ironic tweets were labelled as *other verbal irony.* The category assimilates examples of irony that could not be categorised into *ironic by clash*, nor into *situational irony* and is therefore expected to contain more ambiguous and heterogeneous expressions of irony, which might be more challenging to detect automatically.

To verify the validity of our assumptions, and to get a better understanding of the bottlenecks in irony detection (i.e. whether classification performance depends on the linguistic realisation of irony), we analysed the classification output for the different types of irony in our corpus. Figure 4.2 visualises the accuracy of the classifier for each type of irony and the performance on the two sorts of non-ironic instances (i.e. hashtag corpus versus background corpus). The scores are based on the output of the best-performing combined system (lexical + semantic + syntactic features) and present its recall for the different types of irony in our corpus.

The graph seems to confirm our intuition, showing that the system performs best when detecting ironic tweets that are realised by means of a polarity contrast (i.e. 78% are classified correctly), followed by instances describing situational irony. On the other hand, detecting *other type of irony* appears much more challenging (i.e. recall is 45%). A closer look at *other types of irony* revealed that the instances are often ambiguous and realised in diverse ways, as shown in the following examples. It is important to recall that, prior to classification, the hashtags *#irony, #sarcasm* and *#not* were removed from the tweets.

(28) Trying to eat crackers on the quiet floor likeee.. Maybe if I chew slower no one will notice.. #not

(29) @username hold on a minute. Are you saying All blonde white women look the same?? #sarcasm

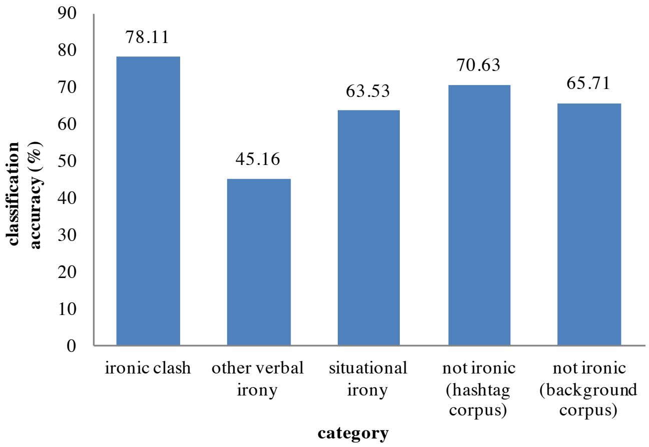


Figure 4.2: Classification accuracy of the best system for the different types of irony.

(30) ‘I like to think of myself as a broken down Justin Bieber’ - my philosophy professor, everyone #sarcasm

Analysis further revealed that they were, as well as instances of situational irony, more often wrongly classified (i.e. *false negatives*) when containing (multiple) hyperlinks and @-replies or mentions. In both cases, information that is crucial to notice irony may be included in an image (inserted with a hyperlink), or in a previous post to which the tweet under investigation is a reply (e.g. example 25).

When looking at the non-ironic instances, we see that the system scores better on non-ironic tweets from the hashtag corpus (i.e. originally containing an irony-related hashtag) than on that from the background corpus (i.e. without such a hashtag). In fact, non-ironic instances from the background corpus were sometimes (very) similar to ironic tweets (example 31), while non-ironic instances from the hashtag corpus often showed larger differences with ironic tweets (examples 32 and 33).

(31) Corny jokes are my absolute favorite

(32) @username Talking to yourself again Mo #Irony

(33) i hate waking up in the mornings 😒 #basicbrianna #not #an #early #bird

This might suggest that tweets that were (erroneously) tagged as ironic by their author differ even more from ironic tweets than non-ironic tweets without such an irony hashtag. This would support the claim that irony hashtags are often used erroneously by the Twitter community (e.g. Hallmann et al. 2016b), implying that irony research imposes manually annotated data. Consequently, an interesting direction for further research would be to explore the classifier's ability to predict such tweets (i.e. non-ironic, although labelled as ironic by the author) correctly if they are not part of the training data.

Finally, classification errors on the *ironic by clash* category include tweets where the irony results from a polarity contrast which cannot be identified using sentiment lexicon features alone (*supra*). We see two possible explanations for this. First, we observed that in the majority (i.e. 77%) of the misclassified tweets, the only clue for a polarity contrast is an **irony-related hashtag** (i.e. #not), which was removed from the data prior to training. In fact, noticing the irony in examples 34 and 35 without such a meta-hashtag would be impossible.

(34) New computer stickers xD The top left one is the best HAHA #irony

(35) Lots of goals tonight in the A League #not

Second, tweets that do not require a meta-hashtag to perceive a polarity contrast, but that were nevertheless missed by the classifier (i.e. 23%), include an **implicit evaluation** or an evaluation as part of a hashtag (e.g. '#ionlygetbetter'). As explained in Chapter 3, understanding such implicit sentiment requires world knowledge rather than sentiment lexicon information (see examples 36 and 37).

(36) Spending the majority of my day in and out of the doctor[NEG] has been awesome. #sarcasm

(37) @username yeah I do. But you know there's this thing called an all nighter[NEG] and apparently I wanna pull one #not

While the oppositions in examples 34 and 35 would be impossible -even for humans- to recognise without such hashtag information, polarity contrast as included in examples 36 and 37 are likely to be identified, on the condition that the system could access common sense or connotative knowledge. As such, it would ideally recognise phrases like 'spending (...) day in and out of the doctor' and 'pulling an all nighter' as related to negative sentiment, and find the contrast with positive opinion expressions such as 'I want' and 'awesome'.

4.6 Summary

In this chapter, we explored the viability of automatic irony detection on Twitter. To this end, a series of binary classification experiments were conducted using a corpus of manually annotated tweets for training and testing. We developed a pipeline to extract different feature types, which enabled us to gain insight into the most contributing information sources for this task. As the classification algorithm we made use of a SVM applied in its default kernel configuration, but we optimised the C and γ hyperparameters in each experimental round.

In short, our system exploits lexical, sentiment, syntactic, and semantic features, several combinations of which were experimentally tested. While similar features are commonly used in the state of the art, we expanded our lexical and semantic feature sets with respectively n-gram probabilities and word cluster information, two features that have insufficiently been explored for this task. To calculate n-gram probabilities, two n-gram ($n= 3$) models were trained on large (ironic and non-ironic) background corpora. Semantic clusters were generated based on word embeddings generated using the Word2Vec algorithm.

Our binary classification experiments exploiting different feature groups revealed that, although lexical, semantic and syntactic features achieved state-of-the-art performance (i.e. yielding F_1-scores between 63% and 67%), none of these feature groups outperformed the character n-gram baseline (68%). This observation is in line with that of Buschmeier et al. (2014) and Riloff et al. (2013), who also reported strong n-gram baselines for irony detection. An explanation might be that the most discriminative features in the lexical feature group are 'undersnowed' by the large number of n-gram and language model probability features. Combining the feature groups, however, results beneficial to the classification performance. Combining lexical with syntactic and semantic information sources yielded an F_1-score of **70.11%**, hereby outperforming the character baseline and lexical features alone with respectively 2 and 3 points. The experiments also demonstrated that combining the output of different models further enhances classification performance by 1.5 point.

Analysis of the system performance on the different types of irony revealed that tweets where the irony results from a **polarity contrast** are more likely to be detected than other types of irony. An important challenge, however, will be the identification of implicit sentiment, i.e. situations that have a prototypical sentiment, such as 'not being able to sleep', 'being ignored', and so on. Furthermore, the analysis also provided insights into the system performance on two types of non-ironic data in our corpus (i) originally carrying an irony hashtag, but annotated as non-ironic and ii) non-ironic tweets from a background corpus) and showed that performance on the former category was better compared to non-ironic tweets from the background corpus.

Although research into irony detection has thrived in recent years, comparison with related research is not trivial, given that many of the discussed papers make use of much larger training corpora (up to 812K tweets), whereas the present study is based on a relatively small dataset (5K). Also important to note when comparing systems is the labelling method of the training data. Many studies make use of hashtag labels (e.g. Bamman and Smith 2015, Davidov et al. 2010, González-Ibáñez et al. 2011), whereas for this study, both training and test corpus were manually annotated according to a fine-grained annotation scheme for irony. Taking into account these nuances, we see that our system compares favourably to the work published by González-Ibáñez et al. (2011) and Riloff et al. (2013), whose experimental setups are the most comparable to that of the present research.

The current experiments are based on information sources that group features according to their nature (e.g. syntactic-, lexical-, semantic- and sentiment-based). In further research, however, it will be interesting to perform feature selection to gain insights into the informativeness of individual features (e.g. specific n-grams, or part-of-speech tags) for this task.

As stated by Joshi, Bhattacharyya and Carman (2016), an important bottleneck in irony detection are situations that carry implicit or prototypical sentiment (i.e. affective information commonly associated with real-world entities or situations, like going to the dentist). While people have access to such world knowledge, machines do not. Analysis of our experiments revealed that, when looking at wrongly classified instances of *ironic by clash*, approximately 20% of the misclassified instances involved implicit sentiment. In the next chapter, we therefore take a closer look at the implicit sentiment expressions (or *targets*) that were annotated in the irony corpus (see Chapter 3), and we take the first steps to detect such implicit or prototypical sentiment automatically.

CHAPTER 5

Modelling connotative knowledge for irony detection

In accordance with irony literature, manual annotations revealed that the irony in our corpus is often realised through a polarity contrast between an explicit and implicit sentiment expression (see Chapter 3). While the former are likely to be recognised using sentiment lexicons, implicit sentiment refers to connotative information associated with natural language concepts, information that is typically not accessible by machines.

In this chapter, we confront the challenge of automatically recognising such implicit sentiment in tweets and hereby provide an answer to (the first part of) our second research question: **'is it feasible to automatically detect implicit or prototypical sentiment related to particular situations?'**. Having at our disposal manually annotated phrases that are linked to their implicit sentiment (e.g. 'working during the weekend', 'feeling sick'), our goal is to develop and evaluate a method to define this implicit sentiment automatically. We propose two methods to tackle this problem, i) by making use of an existing knowledge base and ii) by performing automatic sentiment analysis on crawled tweets (i.e. a data-driven approach).

5.1 Introduction

With Web 2.0, the concept of sharing has acquired a new dimension. Gifted with the ability to contribute actively to web content, people constantly share their ideas and opinions using services like Facebook, Twitter and WhatsApp. Similarly to face-to-face interactions, web users strive for efficient communication, limiting the amount of conversation to what is necessarily required to understand the message, hence leaving obvious things unstated (Cambria et al. 2009). These obvious things can be part of **common sense**: knowledge that people have of the world they live in, and that serves as a basis to form judgements and ideas (Cambria et al. 2009).

While common sense and connotative knowledge come natural to most people, this type of information is not accessible by computers. Perhaps the most salient example of this are sentiment analysis systems, which show good performance on expressions that address sentiments explicitly (e.g. Deriu et al. 2016, Mohammad et al. 2016, Van Hee et al. 2014) like example 38 ('brilliant'), but struggle with implicit sentiment expressions (e.g. 'drains so fast' in example 39).

(38) 3000th tweet dedicated to Andy Carroll and West Ham, brilliant start to the season!

(39) iPhone 6 battery drains so fast since last update and shuts down at 40%.

Such implicit sentiment expressions are devoid of subjective words (e.g. 'brilliant', 'fine', 'overpriced') and rely on commonsense and connotative knowledge shared by the speaker and receiver in an interaction. To be able to grasp such implicit sentiment, sentiment analysers require additional knowledge that provides insights into the world we live in and into the semantics associated with natural language concepts and human behaviour.

While modelling implicit sentiment is still in its infancy (Cambria et al. 2016), such linking of concepts and situations to implicit sentiment will open new perspectives in NLP applications, not only sentiment analysis and irony detection, but also other tasks involving semantic text processing, for instance cyberbullying detection (Dinakar et al. 2012, Van Hee et al. 2015).

State-of-the-art knowledge bases

Although a number of research efforts have been undertaken to transfer commonsense knowledge to machines since the introduction of the concept (e.g. Turing 1950), studies are still scratching the surface of 'common[-]sense computing' (Cambria and Hussain 2015, p. 3). There has been a vast research interest in building semantic and commonsense knowledge bases and exploiting them to develop intelligent systems, including Cyc (Lenat 1995), WordNet (Miller 1995), FrameNet (Fillmore et al. 2003) and DBPedia (Lehmann et al. 2015). The above knowledge bases present structured objective information (e.g. 'the sun is very hot', 'a coat is used for keeping warm') or add semantic links between entries in the form of triples, like <*dentist*> **is_a** <*doctor*>.

For sentiment analysis, however, there is an additional need for knowledge about the typical sentiments people hold towards specific concepts or entities and situations. Initiatives to represent such information include the OMCS *(Open Mind Common Sense)* knowledge base, containing neutral and subjective statements entered by volunteer web users and through a GWAP[1]. ConceptNet (Speer and Havasi 2013) was developed as a framework to represent the statements in OMCS so that they can be computationally processed. Another example is SentiWordNet, in which each WordNet synset (i.e. a set of synonyms) is associated with three numerical scores describing how objective, positive, and negative the terms in the synset are. Finally, SenticNet is a knowledge and *sentics* database (Cambria et al. 2010) aiming to make conceptual and affective information as known by humans more easily accessible to machines. Mainly built upon ConceptNet, the knowledge base contains common sense information for 50,000 concepts and outperforms comparable resources for tasks like sentiment analysis (Cambria et al. 2010).

Research objectives

As the ultimate goal of this thesis is automatic irony detection on Twitter, we investigate the added value of connotative knowledge for this task. The object of study in this chapter are manually annotated phrases that carry implicit sentiment or connotative knowledge (i.e. the feeling a concept generally invokes for a person or a group of people), also referred to as *prototypical sentiment* (Hoste et al. 2016).

In Chapter 3, we observed that many ironic tweets show a polarity contrast between what is said and what is implied, or more specifically: a literal positive evaluation is contrasted by an implicit negative evaluation, or vice versa. In our annotation scheme, such phrases that contain implicit or prototypical sentiment are called *irony targets*. An example is shown in sentence 40, where the explicit positive statement 'I love' is contrasted with the negatively connoted phrases

[1] *Game With a Purpose*: a computer game which integrates human intervention in a computational process in an entertaining way.

'cold winter mornings' and 'my car decides not to start'.

(40) I love[EXP-POS] cold winter mornings[IMP-NEG] when
my car decides not to start[IMP-NEG].

To recognise the irony in such tweets, it is key to identify the words that realise the polarity contrast. As such, two challenges have to be faced. First, one needs to identify sentiment expressions (both implicit and implicit) at various levels of text granularity (words, terms, phrases, etc.). While explicit sentiment expressions (e.g. 'what a terrific meal') mostly contain adjectives, adverbs and verbs, implicit sentiment expressions (e.g. 'my car decides not to start') are much harder to identify. They can be single nouns or verbs, multiword expressions, subordinate clauses (e.g. subject-verb-object sequences), and so on. Second, the polarity of the expressed (or implied) sentiment has to be determined. Explicit sentiment expressions are mostly traceable using a lexicon-based approach. As such, their polarity can be determined using existing sentiment lexicons, which contain a polarity value for each entry (e.g. 'good' → *positive*). A bigger challenge, however, resides in defining the **implicit** polarity of natural language concepts (e.g. 'school', 'rain'), which are either not contained by such lexicon dictionaries, or are tagged with an 'objective' or 'neutral' label.

To tackle this problem, Riloff et al. (2013) take a bootstrapping approach to learn positive and negative situation phrases (as verb phrases) in the vicinity of seed words like 'love', 'enjoy', 'hate', etc. They showed that seed words are useful to find prototypically positive and negative concepts (e.g. 'working' was learned as a negative situation since it often follows 'I love' in ironic text). They found that recognising polarity contrasts using such prototypical sentiment benefits irony detection, but pointed to an important restriction of their approach, namely that only verb phrases were considered as negative situation phrases, and that attached prepositional phrases were not captured. For instance, only 'working' was considered a negative situation in the phrases 'working on my last day of summer' and 'working late 2 days in a row'.

More recent work on the automatic recognition of implicit sentiment has been done by Balahur and Tanev (2016). In fact, the researchers went further than Riloff et al. (2013) in that they tried to model implicit emotions (e.g. joy, disgust, anger, fear) rather than sentiments (i.e. positive or negative). The researchers built EmotiNet, a knowledge base containing 'emotion triggers', or situations that trigger certain emotions based on commonsense knowledge. They made use of Twitter to extract connotative information (e.g. 'failure', 'disease'), and used ConceptNet (Speer and Havasi 2013) to obtain properties of the concepts, with which they subsequently started a new iteration. Although the ultimate goal was to gain insights into implicit emotions in journalistic text (e.g. news reports that are likely to evoke certain emotions in people), by making use of domain-independent resources (i.e. Twitter, ConceptNet), the resulting database could be useful in a broad range of applications.

Both of the above studies take a bootstrapping approach to model implicit sentiment and make use of polar patterns (e.g. 'I love [...]', '[...] makes me sick') to extract an initial set of connoted concepts and situation phrases. In this study, we work the other way around. By starting from manually annotated implicit sentiment phrases (i.e. describing concepts and situations), we avoid having to identify them automatically in text, and we will focus on defining the polarity of these concepts in an automated way. In the present chapter, we therefore aim to answer the following research questions:

- How can we automatically define the implicit sentiment in phrases related to specific concepts and situations?
- How do the results compare to manually annotated implicit sentiment (or connotative knowledge)?
- Could this study lead to a viable method to construct a connotative knowledge base that can be used to enhance automatic irony detection?

While several studies have underlined the importance of implicit sentiment for irony detection (e.g. Giora 1995, Grice 1975, Riloff et al. 2013, Wallace 2015), we present, to our knowledge, the first attempt to model implicit sentiment by using a lexico-semantic knowledge base and a data-driven approach that makes use of real-time Twitter information. Moreover, manual annotations of our irony dataset (see Chapter 3) allowed us to evaluate our approach using gold-standard prototypical sentiment situations (e.g. 'going to the dentist').

5.2 Approach

In the following paragraphs, we explore two methods to tackle this problem. Starting from our manually annotated targets conveying implicit sentiment, we aim to infer this sentiment in an automated way by relying on two different information sources. Firstly, we explore SenticNet, a state-of-the-art commonsense knowledge database which has proven to outperform lexical resources like SentiWordNet (Esuli and Sebastiani 2006) for sentiment analysis tasks (Cambria et al. 2010). Secondly, we make use of Twitter as a primary source of commonsense knowledge. With 328 million active users as of August 2017[2], the microblogging service still ranks among the most popular social networking sites anno 2017, and may therefore provide valuable insights into the general sentiment towards particular events, concepts and situations.

In concrete terms, for each annotated target in our corpus, we aim to infer its implicit sentiment automatically by making use of i) SenticNet polarity values, and ii) by crawling tweets to infer the public opinion related to that target. Both methods will be evaluated against the gold-standard annotations.

As mentioned earlier, the objects of study are the manually annotated targets in our irony corpus (see Chapter 3). Table 5.1 presents a number of example targets and the implicit sentiment that was related to them. In total, 671 unique targets were annotated, 665 of which have a negative connotation, while 6 are positive. This imbalance between positive and negative targets in the dataset is also reflected in the table and confirms earlier findings that irony is more frequently realised by saying something positive while meaning something negative than the other way around (Riloff et al. 2013, Van Hee et al. 2016c).

target	**implicit sentiment**
- working on Christmas	negative
- mondays	negative
- people who lie	negative
- people exercise their freedom of speech	positive
- computer has frozen again	negative
- up all night two nights in a row	negative
- 8 am classes	negative
- when my hair is frozen	negative
- 10/10 score	positive
- long 130km #cycle tomorrow, in the minus degree weather	negative

Table 5.1: Manually annotated implicit sentiment phrases or *targets*.

Thus, the targets describe events, concepts or situations that are considered to have a negative (or positive) connotation. Whether a concept evokes a positive or negative sentiment is a 100% subjective judgment defined by cultural or personal differences. Hence, we are aware that the annotated implicit polarities will not necessarily match the judgment of each individual. Nevertheless, during the annotation procedure, annotators could rely on the context of the tweet to get an impression of the intended sentiment. In case the available context was insufficient, they were asked to judge as generally as possible by

[2] Source: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users

searching additional information when necessary and by prioritising commonly held opinions over their own. For instance, while some people might like, or do not mind, to work on festive days, ‘working on Christmas’ was attributed a negative connotation, assuming that the majority of the public would not like it. An inter-annotator experiment (see Chapter 3) confirmed that, despite the subjective nature of such a task, fairly good agreement (i.e. $\kappa = 0.66$ (round 1) and $\kappa = 0.55$ (round 2)) was obtained.

5.2.1 Using SenticNet to infer implicit sentiment

Introduction

Our first approach to define the implicit sentiment of the targets is a knowledge base approach making use of SenticNet 4 (Cambria et al. 2016). The knowledge base contains denotative (or *semantics*) and connotative (or *sentics*) information associated with 50,000 real-world objects, people, actions, and events. Unlike many other sentiment analysis resources, it contains information about commonsense concepts, instantiated by single words and multiword expressions, such as ‘miss flight’, ‘bake cake’, and ‘celebrate special occasion’. Furthermore, SenticNet was not built by manual labelling of existing resources (e.g. WordNet), but is automatically generated via graph-mining and dimensionality reduction techniques applied to multiple commonsense knowledge sources (Cambria et al. 2010).

The knowledge base is structurally encoded in XML-based RDF-triples and is mainly built upon ConceptNet, the graphic representation of the Open Mind corpus (Speer and Havasi 2013). Its ability to represent polarity values for multiword expressions and implicit sentiment or commonsense knowledge concepts (e.g. ‘exam’, ‘lose temper’) allows it to outperform SentiWordNet for polarity classification tasks (Cambria et al. 2010). Within the framework, **polarity** is defined based on the *Hourglass of Emotions* (Cambria et al. 2010), a classification of emotions into four dimensions, being **pleasantness, attention, sensitivity** and **aptitude**. Activation values for each one of the dimensions relate to a positive or negative polarity for a concept. Semantic information for an entry comprises related concepts or words. Mood tags related to an entry are preceded with a hash sign (‘#’) and were extracted from a large corpus of blog posts that are self-tagged with particular moods and emotions. The tags thus describe a SenticNet concept’s correlation with an emotional state. Figure 5.1 presents an example of SenticNet output for two concepts.

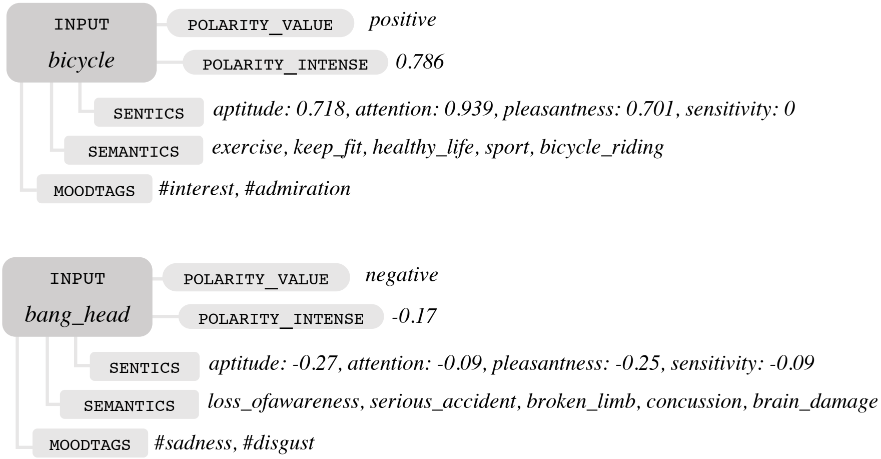

INPUT
bicycle
POLARITY_VALUE
positive
POLARITY_INTENSE
0.786
SENTICS
aptitude: 0.718, attention: 0.939, pleasantness: 0.701, sensitivity: 0
SEMANTICS
exercise, keep_fit, healthy_life, sport, bicycle_riding
MOODTAGS
#interest, #admiration
INPUT
bang_head
POLARITY_VALUE
negative
POLARITY_INTENSE
-0.17
SENTICS
aptitude: -0.27, attention: -0.09, pleasantness: -0.25, sensitivity: -0.09
SEMANTICS
loss_ofawareness, serious_accident, broken_limb, concussion, brain_damage
MOODTAGS
#sadness, #disgust

Figure 5.1: Two examples of SenticNet entries.

Method

For this experiment, we make use of SenticNet 4 and consider its polarity returned for each target as the implicit sentiment related to that target. The knowledge base mainly contains unigrams, bigrams and trigrams, so most targets contain more words than would fit in a single query to the database (see Table 5.1). Consequently, in case the target is a multiword expression or a phrase, we calculated its overall polarity based on the polarities of the individual words or concepts (e.g. 'get_sick') contained by that target. The following paragraphs zoom in on our approach to implicit sentiment modelling using SenticNet 4. Similar approaches have been described by Cambria et al. (2016, 2017) for regular sentiment classification (i.e. finding the polarity of both implicit and explicit sentiment concepts).

As content words, we considered nouns, adjectives, adverbs and verbs, based on the part-of-speech output of the LeTs Preprocess (Van de Kauter et al. 2013) toolkit. This way, polarity values for function words like prepositions and numerals were not taken into account for the global target polarity. In fact, SenticNet provides polarity values associated with words like 'for' (*positive*) and 'two' (*negative*), whose polarity values apply in very specific contexts and are, hence, preferably not taken into account for the global target polarity. To look up multiword expressions, we made use of Rajagopal's concept parser (2013), which makes use of SenticNet as its knowledge base and decomposes the input phrase into commonsense concepts contained by the knowledge base (e.g. 'stomach_flu', 'wear_running_shoes'). The parser breaks sentences into chunks, creates combinations of verb and noun phrases, and searches the best match from a parse graph that maps all the multiword expressions in SenticNet 4. Figure 5.2 visualises such a multiword lookup by means of a flowchart depicting the process from input query to the global target polarity.

Prior to the actual lookup, a number of preprocessing steps were undertaken. Firstly, URLs and @-replies were discarded, since they have no coverage in SenticNet. For the same reason, hash signs ('#') were stripped from hashtag words and punctuation marks were removed. Secondly, concatenated words were split if they were written in camel case (e.g. 'noThanks' → 'no thanks'). Thirdly, common abbreviations were replaced based on an existing dictionary[3], and contractions were expanded (e.g. "would'nt" → would not), since only full forms are included in SenticNet. A next step involved the identification of negation clues and modifiers (Polanyi and Zaenen 2006), as shown in the following examples:

(41) not[NEG] getting any sleep

(42) shouting instructions repeatedly and being completely[INTENS] ignored

The polarity of a sentiment word was flipped when it was preceded by a negation marker (example 41). Whenever a sentiment-bearing word was found in a window of one token to the right of a modifier (i.e. intensifiers or diminishers, see Chapter 3), its polarity was increased (*2) or decreased (*0.5). Next, all targets were tokenised, part-of-speech-tagged, and lemmatised using LeTs preprocess (Van de Kauter et al. 2013) so that lemmas instead of words could be considered for lookup. In a final preprocessing step, all SenticNet queries were lowercased.

Figure 5.2 visualises the automatic sentiment-determining process, starting with the targets as input queries. The queries were preprocessed and broken down into single words or concepts, which were looked up in the knowledge base. In a final step, individual sentiment scores for words or concepts were summed to generate an overall sentiment score for the target.

Analysis and results

After defining the polarity of each target, our method was validated by comparing the labels with the gold-standard annotations. Table 5.2 presents the accuracy of our SenticNet-based polarity assignment by respectively looking at

[3] Source: https://slangit.com

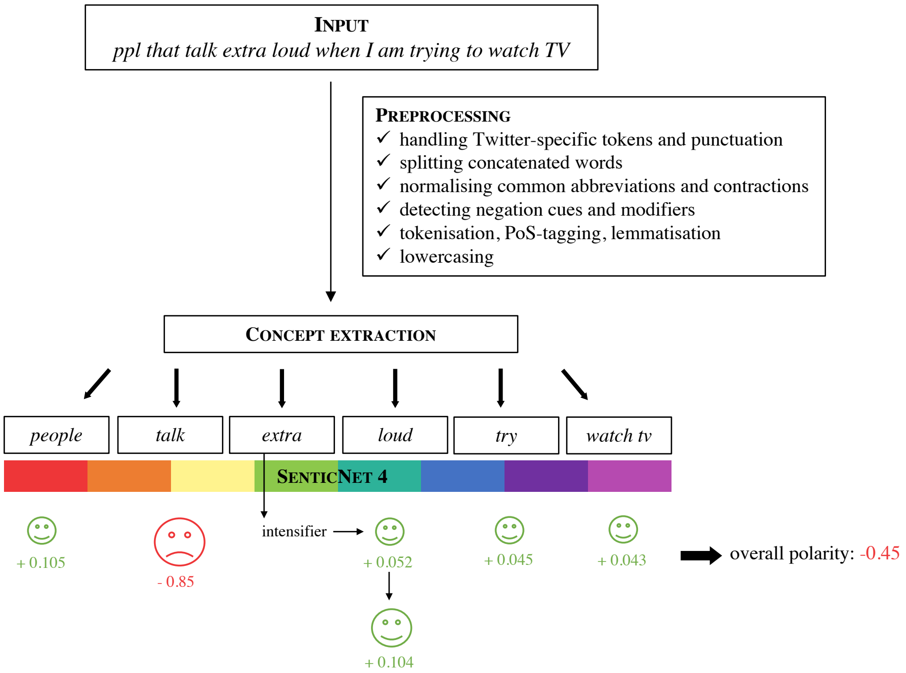

INPUT
ppl that talk extra loud when I am trying to watch TV
PREPROCESSING
✓ handling Twitter-specific tokens and punctuation
✓ splitting concatenated words
✓ normalising common abbreviations and contractions
✓ detecting negation cues and modifiers
✓ tokenisation, PoS-tagging, lemmatisation
✓ lowercasing
CONCEPT EXTRACTION
people
talk
extra
loud
try
watch tv
SENTICNET 4
intensifier
+ 0.105
- 0.85
+ 0.052
+ 0.104
+ 0.045
+ 0.043
overall polarity: -0.45

Figure 5.2: Flowchart visualising concept lookup using SenticNet 4.

all words in the target, only content words, or multiword expressions contained by the target. The same preprocessing steps were each time applied.

	all words	**content words**	**multiwords**
accuracy	33.77%	33.33%	37.25%

Table 5.2: Automatically assigned implicit sentiment using SenticNet 4

Table 5.2 shows that, although connotative knowledge is natural for people, automatically inferring such knowledge is not a trivial task. We observe that searching content words results in a slightly lower score than looking at all words in a target. An explanation would be that the latter takes into account polarity values related to function words, which sometimes might have had a positive effect on the total target polarity. An example is presented below.

(43) **target:** *to feel this hangover*
polarity all words: *feel* (0.72) + *this* (-0.76) + *hangover* (-0.26) = **-0.3**

polarity content words: *feel* (0.72) + *hangover* (-0.26) = **0.46**

Furthermore, we see that the multiword approach yields better results than *all words* and *content words*. This makes intuitive sense, as the former approach protects some 'semantic atoms' (Cambria and Hussain 2015) (if found by the concept parser) which lose their original meaning when broken down into single words. (e.g. 'bang_head').

Effectively, defining a target's valence by summing the polarities of its constituent words or concepts is a rather naive approach, given that the meaning (and hence the associated polarity) of the target depends on the combination of words it contains. Moreover, such an approach cannot resolve contextual ambiguities. Cambria et al. (2017) present an in-depth discussion of this challenge, a clear illustration of which are multiword terms with contrasting constituent words (e.g. 'happy accident' and 'dark chocolate').

Furthermore, specifications can also modify the prior polarity of the word, 'wind', for instance, has a neutral connotation *an sich*, but when combined with '110km/hour', its connotation shifts to negative. Similarly, while 'December' may have a positive connotation for most people, when combined with 'icy roads' or 'electric bills', it becomes negative. Such very specific multiword terms are, however, not contained in SenticNet. In effect, based on the knowledge base, the overall polarity of the latter example would be positive (*december* (+0.799) + *electric_ bill* (−0.04)), although most people would probably agree that the concept evokes a negative sentiment.

Other challenges when using SenticNet to assign a polarity to particular concepts are the lack of coverage of some words (e.g. 'rancid'), and the limited number of inflected forms in the database, since SenticNet stores mainly lemmas and other base forms. Related to the latter, we observed that deriving such base forms from tweets is challenging, given that they often contain misspelled words (e.g. 'your' instead of "you're"), and non-standard abbreviations (e.g. 'fammm' for 'family'). Although we restricted normalisation to the replacement of common abbreviations and normalising contractions to their full form, including a more complex approach to orthographic normalisation (e.g. Schulz et al. 2016) as a preprocessing step could further reduce noise. This is, however, beyond the scope of the current thesis.

Although there is room for optimisation of our SenticNet approach, lack of context, lexical ambiguity, and the inability to perform 'human-like' reasoning with separate concepts will remain an important drawback of this approach. In fact, some phrases or concepts have a negative connotation, although most of the individual words are positively connoted in SenticNet:

(44) Work$^{[-]}$ a double$^{[+]}$ on New$^{[+]}$ Year's$^{[+]}$ Eve$^{[+]}$ and then most of New$^{[+]}$ Year's$^{[+]}$ day$^{[+])}$

(45) Attending$^{[+]}$ a full-day workshop$^{[+]}$ on 2 hours$^{[+]}$ sleep$^{[-]}$

In summary, while knowledge bases like SenticNet present a convenient resource for word- and concept-level sentiment analysis, a more complex approach would be required to define the implicit sentiment of longer sequences or phrases, which often require reasoning or context interpreting. Such understanding of semantic composition involves knowing that people do not like working a double shift, especially not on holidays, and that even the most pleasant activity may become dreadful after a short night's sleep. Still, such a knowledge base would suffer the drawback of its static nature, since even when containing a massive amount of information, it could probably not keep pace with the rapidly evolving world around us, causing commonsense knowledge to be continuously updated.

5.2.2 Crawling Twitter to infer implicit sentiment

Introduction

Collecting myriads of ideas and opinions held by the online community, social media platforms like Twitter present an ideal medium for crowdsourcing or consulting the public opinion. In the following paragraphs, we explore the use of Twitter to automatically define implicit sentiment linked to specific situations and concepts that are subject to irony (or *targets*). In the previous section, we made use of SenticNet 4 to infer implicit sentiment related to particular concepts. We concluded that an important drawback of the method is that polarity information is generally stored at the word (and ocassionally at the multiword) level, while implicit sentiment linked to concepts or situations cannot always be derived from the sentiment linked to their components in isolation.

In this section, we take a machine learning approach to define implicit sentiment based on crawled Twitter data. To be precise, we verify the hypothesis that Twitter provides insights into affective knowledge and investigate whether a large number of explicit opinions about a particular concept or situation are a good indication of the prototypical sentiment related to that concept or situation. In contrast to a knowledge base approach, Twitter imposes few restrictions related to input data. The platform allows the lookup of longer text units and phrases, rather than single- or multiword entries in knowledge bases. Moreover, the medium allows to collect real time opinions held by a large and varied group of people, whereas knowledge bases are generally static and rely on knowledge that has been derived automatically or that has been inserted by experts.

Method

An important first step to infer implicit sentiment using Twitter involves the collection of sufficient tweets for each target, so that a reliable estimation can be made of its prototypical sentiment. We recall that targets are phrases describing connoted situations or concepts (e.g. 'working in the weekend', 'my car won't start'). We made use of the Twitter Search API to collect for each target a set of tweets mentioning that target, determined the prevailing sentiment in these tweets by making use of supervised machine learning (Van Hee et al. 2014).

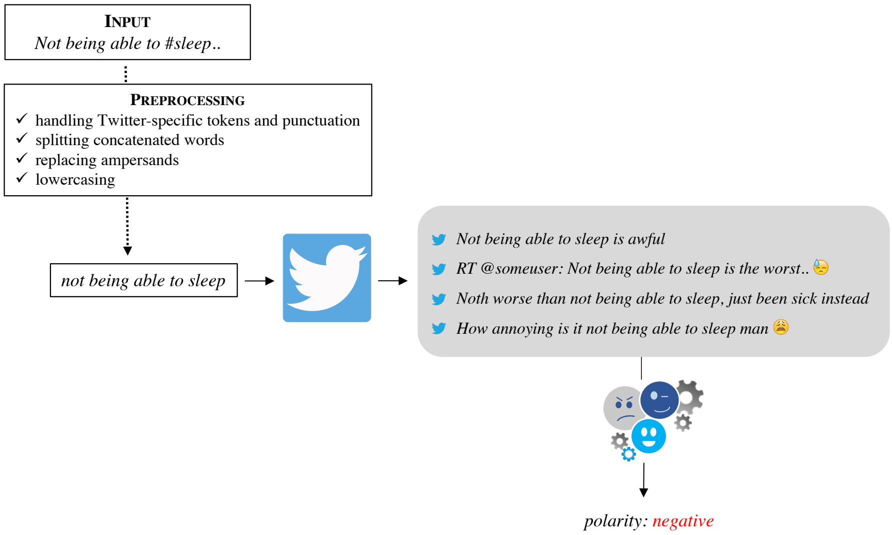
In concrete terms, we applied a state-of-the-art sentiment classifier the architecture of which is described in Chapter 7. In two words, the sentiment analyser is built on a model that was trained on data distributed in the framework of the SemEval-2014 shared task on *Sentiment Analysis in Twitter* (Rosenthal et al. 2014). The sentiment analyser predicts the overall polarity of a tweet as one out of three classes, being *positive*, *negative* and *neutral*. For each target, a Twitter crawl was run to collect the 500 most recent tweets mentioning that target, and sentiment analysis was subsequently applied to predict a sentiment label for each tweet. Next, we calculated the prevailing sentiment in the entire set and considered this the prototypical sentiment associated with that target. As mentioned earlier, the intuition behind this approach is that subjective text (e.g. tweets) about a concept or situation would provide insights into the typical sentiment that this concept evokes, or its connotation. For instance, when a large group of people complain about having to attend lectures at 8 a.m., one could assume it is generally considered an unpleasant activity. Consequently, utterances like 'looking forward to tomorrow's class at 8 am!' are more likely to be ironic. We started with the originally annotated targets as Twitter queries, but explored a number of abstraction methods to see whether these are likely to improve the coverage (see further).

Original annotations as Twitter queries

Each target was used as a Twitter search query. As a first step, all targets were preprocessed to make lookup as effective as possible. The preprocessing steps we describe here are similar to the SenticNet approach (see Section 5.2.1) and include i) handling of Twitter-specific tokens (i.e. URLs and @-replies were removed and hash-signs were stripped off to augment coverage), ii) removal of punctuation marks, iii) splitting of concatenated words in camel case (e.g. 'noThanks' → 'no thanks'), iv) replacement of ampersands (as they have a syntactic function in a search query), and v) lowercasing.

Next, the preprocessed targets were crawled using the Twitter Search API. After collecting a set of tweets for each target, three postprocessing steps involved the removal of duplicates, tweets in which the target did not occur as a consecutive chain, and tweets containing an irony-related hashtag, since we aim to get insights into sincere (i.e. non-ironic) opinions and sentiment related to the targets.

Once a number of tweets for each target were collected and cleaned, we used our sentiment analysis pipeline to define for each tweet whether it was positive, negative or neutral. Based on these predictions, we defined the prototypical sentiment related to each target as the most prevailing sentiment among its tweets. For instance, if 80% of all tweets talking about missing a connecting flight were classified as negative, we considered the prototypical sentiment related to this situation to be negative. The automatically defined implicit sentiment values were then evaluated against the gold-standard labels from the manual annotations.


INPUT
Not being able to #sleep..
PREPROCESSING
✓ handling Twitter-specific tokens and punctuation
✓ splitting concatenated words
✓ replacing ampersands
✓ lowercasing
not being able to sleep
polarity: negative

Not being able to sleep is awful

RT @someuser: Not being able to sleep is the worst..

Noth worse than not being able to sleep, just been sick instead

How annoying is it not being able to sleep man

Figure 5.3: Defining the implicit sentiment of a target using Twitter.

Figure 5.3 visualises the process from preprocessing the target as input query to defining its implicit sentiment based on a set of tweets.

It should be noted that we were only able to crawl tweets for approximately one third of the targets (namely 239 out of 671, 238 of which were negative and 1 positive) when they were looked up in their original form. A possible explanation for the limited coverage is that many targets were too specific to yield many tweets due to containing numbers, personal pronouns, or being vey long (see examples 46 to 48). In fact, analysis revealed that the average length of targets in tokens for which at least one tweet had been found was three, whereas it was nine for targets yielding no tweets.

(46) 7:30 finals on a friday

(47) be the 5th wheel for another New Year's eve

(48) when someone accidentally deletes everything on your phone

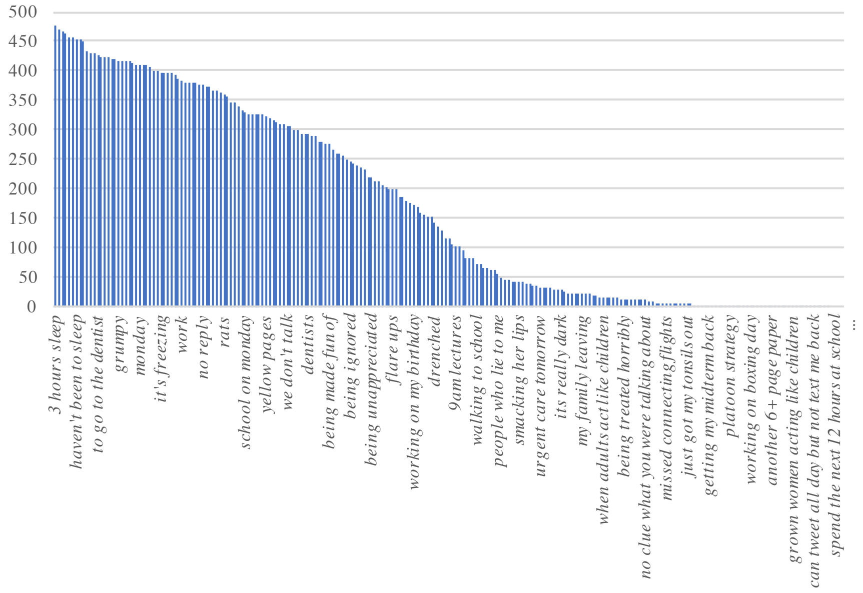


Figure 5.4: Visualisation of the number of tweets crawled for the targets.

Another explanation for the limited coverage of the targets is methodology-related. Using the Twitter Search API does not allow to retrieve historical tweets, but only returns tweets matching the input query from the past seven days. As a consequence, some targets yielded very few or even no results. Figure 5.4 illustrates the number of tweets found for the targets (due to space constraints, only a subset is visible on the x-axis), with the maximum set to 500. After removing duplicates, between 0 and 479 tweets were retrieved per target. The graph shows that the longer and more specific the target, the fewer tweets were found.

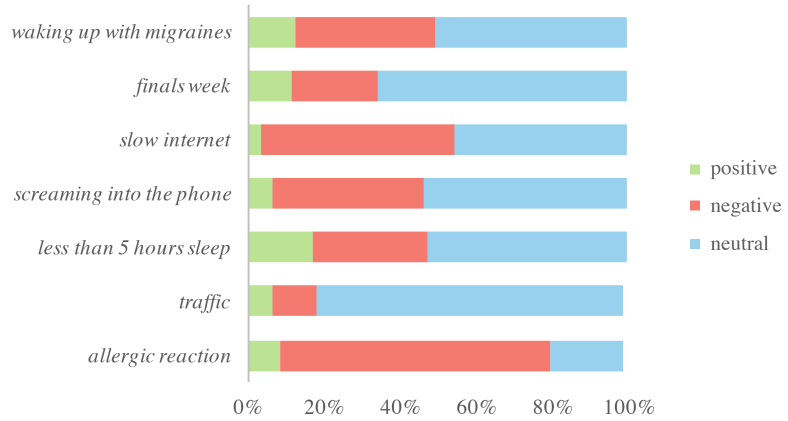


Figure 5.5: Proportion of positive (green), negative (red) and neutral (blue) tweets for a set of targets.

After collecting the tweets, automatic sentiment analysis was applied to determine the prototypical sentiment related to each target. For details on the sentiment analysis pipeline we refer to Chapter 7. Figure 5.5 visualises the sentiment analysis output for a set of example targets: each bar indicates the proportion of positive, negative and neutral tweets for the corresponding target on the y-axis. It can be observed that the opinions expressed towards 'allergic reaction' were mostly negative, whereas they were mostly neutral for 'traffic'.

The graph shows that many neutral tweets were retrieved. In such tweets, either no sentiment, or both positive and negative sentiment are expressed. However, as will be described in the following paragraphs, as we aim to infer connotative information, which is either positive or negative, such tweets are less informative for this task. For this reason, we also defined the overall polarity of a concept without taking neutral tweets into account.

targets	coverage	accuracy (pos/neg/neu)	accuracy (pos/neg)
original	36%	26.78%	71.97%

Table 5.3: Sentiment analysis accuracy after crawling Twitter using the original targets.

Table 5.3 shows the coverage of the original (i.e. as annotated) targets on Twitter and presents the accuracy of the method to define the implicit sentiment for the targets. As can be deduced from the second and third column, when considering all tweet predictions for a given target, the most prevalent sentiment is often neutral, hence resulting in a low accuracy compared to the gold-standard implicit sentiment, which is either positive or negative.

(49) Expert Views: India consumer inflation climbs[NEG] up in March via @username

(50) I took a 3 hour nap after school which means no sleep[NEG] for me...

(51) I had an epiphany. What if I took my energy and put it on all the joyful and positive things in life, rather than on people who lie[NEG] to me?

(52) Working on my birthday[NEG]... guess this is what adulting feels like 😏

As shown in the examples, negative concepts like 'people who lie' and 'no sleep' may occur in neutral tweets (examples 49, 50, 52), or tweets expressing both positive and negative sentiment (example 51).

Re-evaluating our system after discarding such neutral tweets, however, resulted in much better accuracy, namely 72%. This means that for 72% of the targets, we were able to define their implicit sentiment by performing sentiment analysis on a set of tweets mentioning that target. This accuracy also means, however, that for 28% of the targets, the automatically defined sentiment did not correspond to the gold standard. A qualitative analysis revealed that this is due to several causes: i) not all tweets mentioning a negative target (e.g. '9 a.m. lectures') were actually negative (e.g. 'last day of uni 😬 but hey no more 9 a.m. lectures!'), and ii) tweets were sometimes misclassified as they were ironic or ambiguous (e.g. 'yeah I bet you already miss the 9 a.m. lectures').

In sum, Table 5.3 confirms our hypothesis that Twitter data offers insights into the prototypical sentiment related to particular concepts or *targets*. It is important to underline, however, that the results apply to merely 36% of the targets, as we were unable to collect tweets for the remaining 64%. To tackle this problem, the following paragraphs describe a number of strategies to increase the coverage of our targets on Twitter. As shown in Table 5.1, the 671 targets vary greatly in structure and a number of them are very specific. We therefore attempted to convert them into a more abstract and homogeneous list by automatically extracting i) **content words**, ii) **syntactic heads**, and iii) **verb-object (V-O)** patterns.

Content words as Twitter queries

Firstly, we reduced the targets to content words only. Based on Part-of-Speech information obtained using the LeTs Preprocess (Van de Kauter et al. 2013) toolkit, we discarded all words but nouns, adjectives, adverbs and verbs. Other words were replaced by a wildcard (i.e. '*'), meaning that any word could occur at that position, hence allowing a more flexible Twitter lookup.

original target	**content words target**
write psychology papers	write psychology papers
you test my patience	* test * patience
monday mornings	monday mornings
when you say hi to someone in the hallway and they completely ignore you	* * say * * someone * * hallway * * completely ignore *
I work a double on New Year's Eve and then most of New Year's Day	* work * double * new year * eve * then most * new year * day
I have pink eye	* have pink eye
when someone accidentally deletes everything on your phone	* someone accidentally deletes everything * * phone
9 am lectures	* lectures

Table 5.4: Original targets versus content word targets. Function words are replaced by a wildcard.

As shown in Table 5.4, keeping only content words discards pronouns, determiners, etc., and make the targets more likely to yield many tweets. However, it also discards elements that are crucial for the semantics of a target, such as numbers and figures. For instance, keeping only content words, '9 am lectures' becomes '* lectures', which could generate a number of irrelevant tweets as well when used as a search query.

Overall, using content words instead of the original targets provides some abstraction, allowing to collect tweets for 277 out of 671 targets. This is 5% more than when using the original targets as queries. On the downside, more noise is contained in the crawled tweets. Below, some example tweets are presented that correspond to the query '* hour car ride', derived from the target '10 hour car ride'. When comparing examples 53 and 54 to 55 and 56, we see that even detailed information like numerals (e.g. '10 (hours)') is essential for the semantics of a phrase, and hence to its implicit sentiment or connotation.

(53) Glad that my friend who used to live 4 hours away is shortly goin' to be at 1 hour car ride from here 😍

(54) Happy birthday to the only person I could enjoy an 8 hour car ride with!

(55) 4 hour car ride and I forgot my headphones 😭

(56) Well, time for a 10 hour car ride back home... kill me

The above examples not only demonstrate that implicit or prototypical sentiment applies to very specific concepts, but also underline its strongly cultural and personal character.

Table 5.5 shows the coverage and sentiment analysis results for the targets based on content words, again before and after discarding neutral tweets.

targets	coverage	accuracy (pos/neg/neu)	accuracy (pos/neg)
content words	41%	20.94%	72.20%

Table 5.5: Sentiment analysis accuracy based on a Twitter crawl using content word targets.

Although more noise could have been introduced through the use of wildcards, 72.20% of the targets were assigned the correct implicit sentiment, which is slightly better than the accuracy obtained using the original targets (cf. Table 5.3).

Dependency heads as queries

As a second method to make abstraction from the original targets, we made use of dependency information for that target. We considered the head of a dependency relation in a phrase or compound, as it is known to define the core syntactic and semantic properties of its dependents (Poria et al. 2014). A dependency head (e.g. a noun) has generally one or several dependents (e.g. adjectives, possessives, relative clauses) which modify it. We made use of the statistical dependency parser implemented in the Python library spaCy[4], as it has shown to achieve a state-of-the-art performance (Choi et al. 2015). It uses the terms 'head' and 'child' to describe the words connected by a single arc in the dependency tree, representing a syntactic relation that connects the child to its head.

It is important to note that, after extracting the dependency heads of each target, we decided to re-insert two elements to reduce the loss of crucial semantic information: i) negation words (i.e. 'not') and ii) words that form a compound with a head (e.g. 'psychology papers' was tagged as a compound by the dependency parser, hence 'psychology' was preserved, in addition to 'papers'). Table 5.6 presents some example targets for which we extracted dependency

[4] http://spacy.io

heads. Similarly to the content-words approach (cf. Table 5.4), words that had been discarded were replaced by wildcards ('*').

original target	**dependency heads**
write psychology papers	write psychology papers
you test my patience	* test * patience
monday mornings	monday mornings
when you say hi to someone in the hallway and they completely ignore you	* * say * to someone in * hallway * * * ignore *
I work a double on New Year's Eve and then most of New Year's Day	* work * double on * year * eve * * most of * year * day
I have pink eye	* have * eye
when someone accidentally deletes everything on your phone	* * * deletes everything on * phone
9 am lectures	* am lectures

Table 5.6: Original targets versus dependency heads in the targets.

Using dependency heads instead of the original targets allowed to collect tweets for 347 out of the 671 targets (i.e. 52%). Hence with this approach, Twitter coverage is higher compared to the original targets or content word targets. However, making the targets more abstract also means information loss and a potential change in semantics (e.g. '* am lectures' instead of '9 am lectures' and 'have * eye' instead of 'have pink eye').

Table 5.7 shows the sentiment analysis results for the tweets that were crawled using dependency heads in our targets as queries.

targets	**coverage**	**accuracy (pos/neg/neu)**	**accuracy (pos/neg)**
content words	52%	19.22%	72.07%

Table 5.7: Sentiment analysis accuracy based on a Twitter crawl using dependency heads as queries.

Similarly to the two other approaches, most of the targets were predicted as neutral, yielding an accuracy of 19%. This is similar to the score obtained with content words and would suggest that, the more general the query, the higher the likelihood of retrieving neutral tweets, or tweets with a combination of positive and negative sentiment. When discarding the neutral tweets, however, sentiment analysis accuracy increased to 72.07%.

Verb-object patterns as queries

Finally, we made abstraction by extracting **verb-object (VO)** patterns from the targets. As stated by Riloff et al. (2013), verb phrases are typical structures for negative situation phrases that are common in ironic tweets. Table 5.8 presents some example targets and the verb-object patterns that were extracted. Evidently, no such patterns could be derived for targets consisting of a noun phrase, for instance, which are indicated by 'n.a.' in the table.

original target	**V-O pattern**
write psychology papers	write papers
you test my patience	test patience
monday mornings	n.a.
when you say hi to someone in the hallway and they completely ignore you	say hi, ignore you
I work a double on New Year's Eve and then most of New Year's Day	work double
I have pink eye	have eye
when someone accidentally deletes everything on your phone	deletes everything
9 am lectures	n.a.
Christmas shopping on 2hrs sleep	n.a.
8.30am conference calls	n.a.
DC rush hour	n.a.

Table 5.8: Verb-object patterns of targets.

As illustrated in Table 5.8, negative situation phrases are not exclusively composed by verb-object sequences. In fact, the method allowed to collect tweets for 312 out of the 671 targets (i.e. 47%), which is more than obtained with the original targets and content word targets, but less than the dependency heads approach.

Table 5.9 presents the coverage and sentiment analysis results obtained using verb-object sequences in our targets as queries. If more than one verb-object phrase had been extracted from a target, we considered the predicted sentiment of all phrases in that target (i.e. 'positive' if all V-O strings were positive, 'negative' if all were negative, 'neutral' if one or more were positive and one or more others were negative).

As can be deduced from the table, sentiment analysis performance is slightly lower compared to the other approaches (i.e. original targets, content words,

targets	**coverage**	**accuracy (pos/neg/neu)**	**accuracy (pos/neg)**
V-O patterns	46.50%	17.68%	68.17%

Table 5.9: Sentiment analysis accuracy based on a Twitter crawl using dependency heads as queries.

dependency heads). An explanation is that extracting verb-object patterns from concepts or situation phrases implies loss of information. In cases where this does not affect the semantics too much, such abstraction may be desirable (e.g. 'write papers', 'test patience', 'ignore you'), but in other cases, the discarded information might be essential to the target's semantics and therefore, to its implicit sentiment. For instance, reducing the phrase 'taking the subway alone at 2:40 a.m.' to 'taking subway' discards the element that gives it a negative connotation (i.e. 'alone at 2:40 a.m.'). In other examples, keeping only verb-object patterns implies that the implicit sentiment of the original target becomes less strong (e.g. "work a double on New Year's Eve" → 'work double'). Also, Table 5.8 suggests that considering V-O patterns alone as expressions of implicit sentiment (cf. Riloff et al. 2013) is a too restricted approach, since many implicit sentiment expressions contain noun phrases as well (e.g. '9am lectures', 'DC rush hour', etc.).

Results and analysis

In the previous paragraphs, we explored four methods to crawl tweets for a set of implicit sentiment phrases (or *targets*) that were manually annotated. We automatically determined the prototypical sentiment related to a target by applying sentiment analysis to a set of crawled tweets and to define the prevailing sentiment in this set of tweets.

We can conclude that applying sentiment analysis to relevant tweets is a viable method to define the prototypical sentiment related to a particular concept (yielding an accuracy of up to 72.20%). However, our targets being very specific and restricted by the limited search space when using the Twitter Search API, we were only able to collect tweets for 36% of the targets. Extracting content words, dependency heads and verb-object patterns from the targets as strategies to improve coverage allowed to collect tweets for respectively 43%, 52% and 47% of the targets, hereby outperforming the coverage of the original targets. Analysis of these approaches revealed that, while some information can be discarded without meaning loss (e.g. pronouns, determiners), removing other elements (e.g. numerals) does imply a loss of or change in meaning (e.g. '10 hour car drive' versus 'hour car drive').

When looking at the sentiment analysis results, we see that the methods perform comparably, although it should be noted that the original targets yielded less neutral tweets, probably because the tweets are more specific and hence more likely to reveal an unambiguous sentiment. This was confirmed by a qualitative analysis, showing that the content word, dependency head and V-O queries resulted in a noisier set of tweets than the original targets.

As such, although coverage is rather low (36%) when using the original targets as Twitter queries, these yielded tweets that are semantically the closest to the original targets and are therefore more likely to reflect the prototypical sentiment related to these targets.

For practical motivations, we defined a maximum of 500 search results when crawling the targets using the Twitter API. Given that most targets were very specific, and the API restrictions imply that no historical results can be returned, most targets did not even yield this maximum of 500 tweets. We wanted to investigate, however, whether sentiment accuracy increases with the number of tweets returned for a target. One could hypothesise that, the larger a set of tweets available for a particular target, the more likely it is that the tweets form a good representation of the public opinion and hence, prototypical sentiment related to the target.

We tested this hypothesis with 34 targets for which we were able to collect 2,000 tweets. We automatically determined the implicit sentiment using an incremental number of tweets and plotted the results, as shown in Figure 5.6.

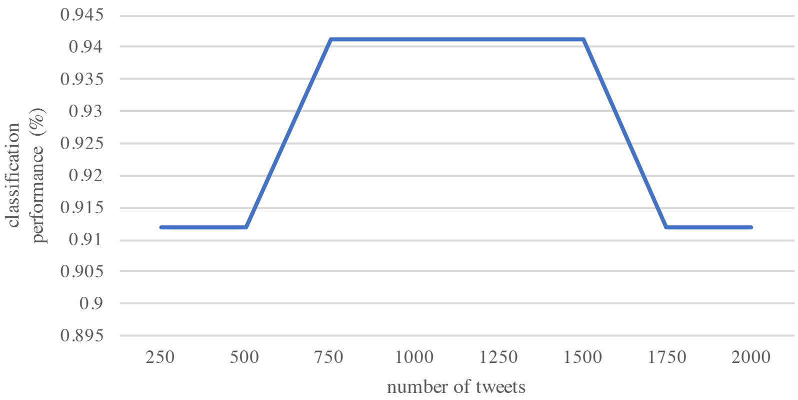


Figure 5.6: Sentiment analysis results by gradually incrementing the number of tweets.

As can be inferred from the above figure, collecting more tweets seems to have a moderate effect on the overall sentiment analysis performance (91% *versus* 94%). However, the increase seems to stagnate at 750 tweets and the scores even diminish as the number of tweets further increases. This would suggest that using more tweets to determine the prototypical sentiment related to a concept does not necessarily give a better indication of that prototypical sentiment. One reason could be that collecting more tweets also could mean more irrelevant tweets or *noise*. The effect is, however, measured on a very small sample (i.e. 34 targets), so the results should be interpreted carefully.

5.3 Summary

In this section, we investigated the feasibility to automatically define implicit or prototypical sentiment related to particular concepts and situations (i.e. *targets*). We started from the manually annotated targets in our corpus and investigated the feasibility to infer implicit sentiment automatically by making use of SenticNet 4 and Twitter. The following two paragraphs provide an answer to our first two research questions in the introduction of the chapter.

Experiments using SenticNet 4 revealed that prototypical sentiment seldom applies to words or phrases in isolation. In fact, many concepts are non-compositional, hence, their meaning cannot be derived from the meaning of their components in isolation. In effect, looking up individual words or even multiwords contained by a target turned out in being too naive an approach to define the implicit sentiment for that target, as the approach breaks semantic atoms down into single words that consequently lose their prior meaning. Moreover, the semantics of a word, and its connotation, often depend on the combination of words it is surrounded by, which is not taken into account by a knowledge base approach either. Finally, we concluded that even a very large knowledge base would probably not be able to keep up with the rapidly evolving world in which commonsense and affective knowledge are constantly updated.

Our second approach underscored that user-generated content such as tweets provides valuable insights into the general or prototypical sentiment related to particular concepts or situations. Our experiments demonstrated that collecting and automatically determining the sentiment of a collection of tweets about a concept or situation proves a viable method to determine its prototypical sentiment. Using the original targets, as well as a more abstract version of the targets (i.e. based on content words, dependency heads and V-O-patterns) both proved to be good methods to collect Twitter data for sentiment analysis, although the latter abstraction method would benefit from including noun phrases and numerals. In effect, the experiments revealed that only half of the targets show a verb-object pattern.

An important advantage of using Twitter to infer connotative knowledge, as compared to SenticNet, is that it allows to look up longer phrases and hence integrate context when defining implicit sentiment. Moreover, it presents a method to consult the public opinion in real time about topical concepts before these could even be inserted in knowledge bases. Two drawbacks of the method are, firstly, that it is more complex than a knowledge base lookup as it requires a (sufficiently large) set of relevant tweets about a particular concept, and a well-performing sentiment classifier to determine the prevailing sentiment in this set of tweets. Secondly, the prototypical sentiment of a situation being based on real-time opinions, it might be influenced by crises or trends, which can cause fluctuations in the public opinion towards a specific concept or situation.

To answer our third research question in this chapter, namely ‘Could this study lead to a viable method for constructing a connotative knowledge base?’, we conclude that the proposed method provides valuable insights into the implicit sentiment related to a set of concepts. We subsequently showed that choosing one method to convert the implicit sentiment phrases into more abstract or more general Twitter queries is not trivial, given that the semantically important information depends from one concept to another (e.g. ‘10’ is crucial information in ‘10 hours car ride’, whereas it is not in ‘not being able to sleep 10 nights in a row’. In the latter example, not being able to sleep one night may already be considered negative). An interesting direction for future research will therefore be to explore more specific patterns of implicit sentiment expressions, for instance by analysing Part-of-Speech sequences.

CHAPTER 6

Automatic irony detection based on a polarity contrast

In the previous chapter, we explored two methods to automatically define the implicit or prototypical sentiment related to connoted concepts and situations, which is an essential step to identifying polarity contrasts in ironic text. For this purpose, we made use of the common sense information stored in SenticNet 4 and automatically derived sentiment from crawled Twitter data.

In this chapter, we aim to answer the second part of our second research question, being **"does our approach [to recognise implicit sentiment automatically] benefit automatic irony detection?"**. To this purpose, we will evaluate the performance of a clash detection system for irony recognition, and investigate its added value for the irony detection system described in Chapter 4.

6.1 Introduction

As mentioned earlier in this thesis, manual annotations revealed that about 70% of all ironic tweets in our corpus contain a polarity contrast (see Chapter 3). Also, in about half of these tweets, the contrast between the literal and the intended polarity can only be perceived through the presence of an irony-related hashtag. In example 57, the hashtag *#not* inverts the polarity of the expression 'we all enjoyed'.

(57) We all enjoyed today's workshop #not

Other ironic tweets, however, do not require such a hashtag. The polarity contrast is realised in the text itself, either by two explicit evaluations, or by an explicit and implicit evaluation (i.e. a phrase with prototypical sentiment). Below, we recapitulate examples 11 and 12 (see Chapter 3) to illustrate this:

(58) I just love[EXP-POS] when you test my patience[IMP-NEG]! 😌 #Not

(59) Sitting in this hall is fun!![EXP-POS] 😫[EXP-NEG]

In Chapter 5, we showed that analysing opinions expressed by the 'Twitter crowd' is a good strategy to infer implicit sentiment or connotative knowledge related to specific concepts or situations. In short, we applied automatic sentiment analysis to a set of crawled tweets about connoted concepts or situations (e.g. 'going to the dentist', 'finals week'), and considered the prevailing sentiment in this set of tweets as the prototypical sentiment related to that concept or situation, with succes.

In this chapter, we will combine this method with the identification of explicit subjective words to define whether a polarity contrast is present in a tweet. Previous work on irony detection has investigated the added value of explicit or implicit sentiment contrast information (e.g. Barbieri and Saggion 2014, Peng et al. 2015, Riloff et al. 2013). Other studies have focussed on modelling implicit sentiment or building connotative knowledge bases (Balahur et al. 2011, Balahur and Tanev 2016, Cambria et al. 2016, Riloff et al. 2013, Singh et al. 2002). However, we present, to our knowledge, the first approach to include explicit and implicit polarity contrast information for irony detection based on real time prototypical sentiment that is automatically extracted from Twitter.

First, it should be noted that a contrast feature (taking only explicit polarity information into account) was already included as part of the sentiment lexicon features (see Chapter 4). However, as experiments revealed that combining lexical, semantic and syntactic features obtained the best results in our irony detection experiments, the feature was not included in the SVM-based approach we refer to in the current chapter. In this chapter, we aim to re-evaluate our SVM-based irony classifier (see Chapter 4 for a thorough description) after taking polarity contrast information into account.

In short, the SVM classifier is informed with polarity contrast information in two ways:

1. by means of a **binary feature** indicating the presence of a polarity contrast (i.e. between explicit and explicit and/or explicit and implicit polarities) in a tweet (with *1* meaning 'yes', *0* meaning 'no');
2. as a class label for irony (i.e. if a polarity contrast is present, the tweet is ironic, otherwise it is not) besides the SVM classifier. Both predictions are combined for the final irony prediction of a tweet.

The latter (combined) system is applied in two ways: a tweet is predicted as ironic if i) the SVM and clash-based system agree that it is ironic and ii) if one of the two systems predicts it as ironic. As we have at our disposal gold-standard (i.e. manually annotated) implicit sentiment information, we will evaluate the performance of the polarity clash system based on gold-standard and automatically defined implicit sentiment.

System evaluation

To facilitate reading, all results in this chapter are reported on the testdata (see Chapter 4 for details on the experimental corpus), while the experimental setups and evaluation methods were defined by experimenting on the training data. The contrast-based approach to irony detection will be implemented and evaluated in different ways throughout this chapter. Firstly, in ironic tweets, an explicit evaluation can be contrasted with i) another explicit evaluation as shown in example 59 (it is denoted as [exp-exp] in the tables) or ii) an implied evaluation as in example 58 ([imp-exp] in the tables). The system is evaluated in two flavours: firstly, a polarity contrast is identified in either one of the two situations and secondly, only polarity contrasts as in the latter situation (i.e. an explicit evaluation is contrasted with an implied evaluation) are taken into account. We do this for two reasons. As irony is mostly realised implicitly (e.g. Giora 1995), it may be a stronger indicator for irony than explicit polarity contrasts. Furthermore, given the subjective nature of Twitter, considering contrasting polarities in a tweet as a clue for irony might cause overgeneration. Finally, in case of an [imp-exp] clash, the polarity contrast will be defined using both gold-standard and automatically recognised implicit sentiment.

After including the polarity contrast feature, we evaluated the SVM classifier in the same way, namely by looking at two different realisations of a polarity contrast, and taking into account gold-standard and automatically defined implicit sentiment.

When interpreting the results, it is important to recall, however, that not all ironic instances in our corpus are realised through such a polarity contrast. We review the distribution of the test data in Figure 6.1. Out of the 958 instances, 485 are ironic, while 473 are not. The ironic category is composed by *ironic by clash, situational irony* and *other irony*. From the first category, 171 (51%) require an irony-related hashtag to perceive the polarity contrast.

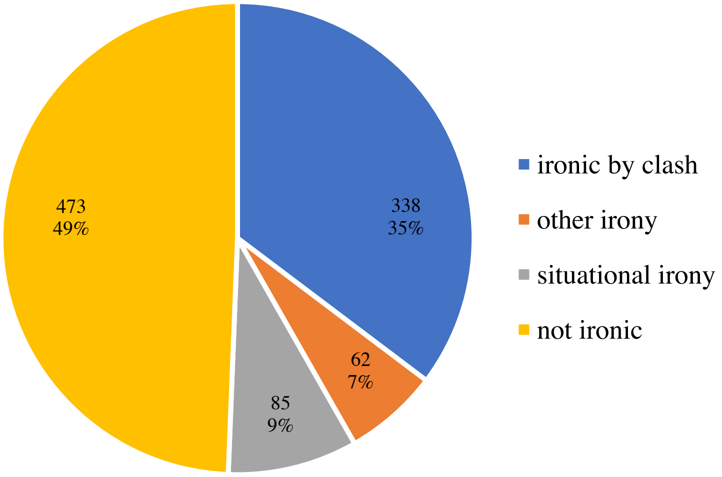


Figure 6.1: Distribution of the test data.

As such, besides calculating the overall system performance (i.e. including all types of irony in our corpus), which allows to compare the results with our SVM classifier described in Chapter 4, we also investigate its performance on *ironic by clash* instances. Moreover, since half of these *ironic by clash* tweets cannot be recognised as such without the presence of an irony-related hashtag, it also makes sense to calculate the performance of the system for the instances where no such a hashtag is required (see examples 58 and 59).

Furthermore, we will compare the performance of our approach to that of Riloff et al. (2013), as they adopted a very similar method. By making use of bootstrapped learning, they extracted positive sentiment and negative situation phrases from hashtag-labelled ironic tweets and used the resulting lexicons in a contrast-based method to irony detection. In a next step, the system was combined with an *n*-gram based SVM classifier to evaluate how both systems complement each other.

6.2 A polarity clash-based approach

As a first step, we explore the potential of an automatic contrast detection method to recognise irony, without combining it with our SVM classifier (cf. Chapter 4). The design is straightforward: if the system identifies a polarity contrast, the corresponding tweet is predicted as ironic, otherwise, it is considered non-ironic. We employ the Twitter-based method (using content words as search queries) as described in Chapter 5 to define the implicit sentiment related to specific situations and phrases, and make use of the sentiment lexicons described in Chapter 4 to identify explicit sentiment expressions. More precisely, to search for explicit polarity words that contrast with the implicit sentiment expressed in the tweet, we search for explicit polarity clues in the remainder of the tweet (i.e. text that is not part of the implicit sentiment expression) by using the sentiment lexicons described in Chapter 4. Negation cues are taken into account by flipping the polarity of a sentiment-lexicon match if the word is preceded by a negation word (e.g. *not, no, none*, etc.).

As such, a polarity contrast was found if the tweet contains a contrast between either two explicit evaluations (e.g. 'Lovely morning ... #hate rain') , or a contrast between an explicit and implicit evaluation (e.g. 'Cannot wait to go to the dentist tomorrow!').

6.2.1 Gold-standard implicit sentiment

We start by evaluating the performance of the clash-based system when incorporating gold-standard implicit sentiment information. This will provide insights into the practical feasibility of the approach, while leaving aside the challenge of automatically identifying implicit sentiment.

The system automatically identifies explicit sentiment expressions in each tweet and predicts the tweet as ironic if the explicit sentiment contrasts with that of the implicit sentiment. In concrete terms, given the tweet in example 60 (note that the hashtag *#not* is left out), the system is provided with the negative implicit sentiment related to 'you test my patience'.

(60) I just love when you test my patience! 😌 #Not

By making use of existing sentiment lexicons, it will then look for explicit polarity expressions that contrast with the negative implicit sentiment (e.g. 'I just love' and '😌'). More precisely, by relying on the sentiment lexicons as described in Chapter 4, the system identifies positive (e.g. ‘tolerant’, ‘funny’) and negative (e.g. ‘worthless’, ‘painful’) polarity words.

Results

Table 6.1 presents the performance of the system on the testdata (see Chapter 4). As explained earlier, two implementations of a polarity contrast were evaluated: i) an explicit sentiment contrasted with either an implicit or another explicit sentiment ([imp-exp] or [exp-exp]), and ii) an explicit sentiment contrasted with an implied sentiment ([imp-exp]). Recall, precision and F_1-score are calculated on the positive class instances.

	positive class	**clash**	**accuracy**	**precision**	**recall**	**F_1**
1	ironic by clash + situational + other	[imp-exp] or [exp-exp]	56.99%	57.78%	55.88%	56.81%
2	ironic by clash + situational + other	[imp-exp]	61.80%	100%	24.54%	39.40%

Table 6.1: Performance of the clash-based system for irony detection using gold-standard implicit sentiment information.

When evaluating the system on the entire positive class (i.e. *ironic by clash + situational irony + other irony*), we observe that the clash-based system does not outperform the optimal SVM classifier as described in Chapter 4 (F_1= 56.81% and F_1= 39.40% versus F_1= 70.11%, respectively). This can be explained by the fact that the former is targeted towards instances where the irony results from a polarity contrast, which constitute ‘only’ 70% of the irony class (see Figure 6.1). Moreover, it has been shown that, in about 50% of the ironic instances, an irony-related hashtag is required to notice the irony, as shown in the following examples:

(61) wtf my english.. so perf in the morning ✌ #not

(62) I’m sure i aced this quiz ... lol #sarcasm

Such instances are missed by the clash-based system, while the SVM classifier might pick up other clues that are indicative of irony (e.g. punctuation, interjections, etc.). A qualitative analysis further revealed that the system in setup 1 tends to overgenerate, as it also predicts non-ironic instances as ironic whenever several polarity words are identified in the tweet (see examples 63 and 64).

(63) one of my favorite$^{(+)}$ things to do is to make people wonder$^{(+)}$ ...it was

completely unintentional$^{(-)}$ but buying bogs mud$^{(-)}$ boots in a dress will do it

(64) work hard$^{(-)}$ in silence ; let success$^{(+)}$ make the noise$^{(-)}$.

This supports the hypothesis we formulated in the introduction of this chapter that, due to the subjective nature of Twitter, the presence of an explicit polarity contrast as an indicator for irony might cause overgeneration of the model.

Although the clash-based system does not outperform the SVM classifier on all ironic instances (i.e. *ironic by clash, other irony, situational irony*), it is worthwhile to note that the former is able to recognise ironic instances (i.e. 58 to be precise) that the SVM classifier overlooks, including examples 65 and 66, where a polarity contrast is realised between explicit and implicit sentiment (setup 2):

(65) Spending the majority of my day in and out of the doctor$^{(-)}$ has been awesome$^{(+)}$.

(66) Literally half of the finals i have this semester are today$^{(-)}$, and that's totally not stressful$^{(+)}$ at all!

Given that the contrast-based method is targeted towards instances of irony that include a polarity clash, it makes sense to calculate its performance on these specific instances in the test corpus. Moreover, we evaluate its performance on instances where the polarity clash should be clear from the text (i.e. no irony hashtag is required). Table 6.2 presents the recall of the system (i.e. how many instances of *ironic by clash* were correctly labelled by the system) for both categories. It is important to note that only recall can be reported here. Because the system only outputs binary labels, we cannot distinguish the different subcategories. As a result, we cannot count the number of false positives for one subcategory. We can, however, count the number of false negatives per subcategory and calculate recall. We then assume that each positive prediction is also a true positive for the subcategory in question.

	positive class	**clash**	**recall**
3	ironic by clash	[imp-exp] or [exp-exp]	56.51%
4	ironic by clash	[imp-exp]	35.21%
5	ironic by clash (no hashtag required)	[imp-exp] or [exp-exp]	83.23%
6	ironic by clash (no hashtag required)	[imp-exp]	71.26%

Table 6.2: Performance of the clash-based system for irony detection on *ironic by clash* using gold-standard implicit sentiment information.

While Table 6.2 shows results of the system for the task it would actually be designed for (i.e. identifying ironic instances that are realised through a polarity contrast), we see that recall is still not very high. However, it can be deduced from the table that many false negatives include instances where an irony hashtag is required to understand the irony, as recall increases considerably in 5 and 6, compared to setups 3 and 4. These are, in fact, instances that require additional context to understand the irony. Without such additional context, it is impossible -even for humans- to detect irony in examples like sentence 67:

(67) @username Yes it was a GREAT party!

A qualitative analysis revealed that false negatives in 5 and 6 include instances for which the system was unable to identify the explicit sentiment expression that contrasts the implicit sentiment information in the tweet. Examples of the difficulties here include concatenated (e.g. ‘#excitingtimes’) or noisy (e.g. ‘yaayyyyy’) sentiment expressions, and phrases like ‘[...] is exactly what I need’, (example 68), whose polarity cannot be inferred using a sentiment lexicon-based approach.

(68) Because a field trip two days before finals is exactly what i need

In sum, the results demonstrate that the presence of a polarity contrast is a strong indicator for irony, at least if no hashtag is required to notice the irony. As shown in Tables 6.1 and 6.2, identifying a polarity contrast between [imp-exp] or [exp-exp] benefits recall of the system, although it is prone to overgeneration, as subjective words are not exclusively present in ironic tweets. Less prone to overgeneration are polarity contrasts between explicit and implicit sentiment, which have shown to yield very high precision (see Table 6.1) if the system is informed with perfect (i.e. gold standard) implicit sentiment information. Overall, the clash-based system does not outperform the SVM-based approach to detect irony, but it does, however, detect a number of ironic tweets (i.e. 58) that contain a polarity contrast and that were overlooked by the latter approach.

6.2.2 Automatic implicit sentiment

In the previous section, we investigated the performance of a clash-based irony detection system when gold-standard implicit sentiment information was used. In this section, we explore the performance of the system when such implicit sentiment is defined automatically. In concrete terms, for each tweet where an implicit sentiment or connoted situation phrase (e.g. ‘being a light sleeper’) was annotated (see Chapter 3), we inferred the polarity of this implied sentiment automatically by making use of Twitter (see Chapter 5).

In concrete terms, for each tweet containing a connoted sentiment phrase, we extracted content-word patterns (since they would have more coverage on Twitter than the original phrases, without much meaning loss (see Chapter 5)), for instance 'being a light sleeper' → 'being * light sleeper'. As a next step, we collected tweets for the content-words patterns, and subsequently applied automatic sentiment analysis to infer the overall sentiment in the tweets, and hence the prototypical sentiment related to the phrase. Once this had been done, similar steps were taken as in the previous section to recognise explicit sentiment expressions and determine whether a polarity contrast was present in a tweet.

Results

As can be deduced from Table 6.3, the system performance shows a substantial drop compared to the system based on gold-standard implicit sentiment, especially in the second setup. A qualitative analysis revealed that the main reason for this drop is that we were unable to collect tweets for about half of the implicit sentiment expressions, hence their implicit sentiment could not be determined.

	positive class	**clash**	**accuracy**	**precision**	**recall**	**F$_1$**
1	ironic by clash + situational + other	[imp-exp] or [exp-exp]	51.88%	52.86%	45.77%	49.06%
2	ironic by clash + situational + other	[imp-exp]	55.11%	100%	11.34%	20.37%

Table 6.3: Performance of the clash-based system for irony detection using automatic implicit sentiment information.

The same tendency can be observed when looking at the results for the category *ironic by clash* in Table 6.4.

	positive class	**clash**	**recall**
3	ironic by clash	[imp-exp] or [exp-exp]	42.01%
4	ironic by clash	[imp-exp]	16.27%
5	ironic by clash (no hashtag required)	[imp-exp] or [exp-exp]	53.89%
6	ironic by clash (no hashtag required)	[imp-exp]	32.93%

Table 6.4: Performance of the clash-based system for irony detection on *ironic by clash* using automatic implicit sentiment information.

The scores are much lower in setup 4 and 6 compared to 3 and 5. An explanation for this observation is that instances that were missed by the systems 2 and 4 were found by the systems 1 and 3 if they contained words with opposite polarities, as illustrated in example 69:

(69) so ill it actually hurts to breathe. still that nice long walk to the station did me the world of good #not

In the tweet, the implicit sentiment related to ‘so ill it actually hurts to breathe’ could not be defined automatically. An explicit polarity contrast was perceived, however, between the words ‘hurts’ and ‘long’, and ‘nice’ and ‘good’. As a result, the tweet was predicted as ironic.

Overall, we can conclude that the presence of a polarity contrast is a strong indicator for irony, on the condition that implicit sentiment expressions can accurately be detected. To determine implicit or prototypical sentiment, we made use of Twitter, as it has shown to provide for a reliable method for this task (cf. Chapter 5).

6.3 Combining an SVM with polarity contrast information

In this section, we evaluate the performance of the clash-based system for irony detection when combined with our SVM-based approach (see Chapter 4). As explained in the introduction of this chapter, we combined the information provided by the two classifiers in two ways, firstly by including the output of the contrast-based method as a feature for the SVM classifier, and secondly by combining the output of both systems.

6.3.1 SVM exploiting a polarity contrast feature

In short, the output of the polarity-contrast system was added as a binary value (i.e. 1/0 if a polarity contrast was present/absent in the tweet) to the feature space of the SVM classifier. Next, the model was retrained and evaluated on the test corpus. Like in the previous section, evaluation of the system is done by considering gold-standard and automatically defined implicit sentiment. It is important to note that, based on the results on our training corpus, the contrast feature was activated if either a polarity contrast between two explicit evaluations or between an explicit and implicit evaluation was observed ([imp-exp] or [exp-exp]).

The results of the experiments are presented in Tables 6.5 and 6.6, the former of which presents the results of a cross-validated grid search on the training data to define optimised hyperparameter settings for our SVM classifier.

system	**positive class**	**implicit sentiment**	**optimised parameters**	**cross-validated accuracy**
SVM+clash feat.	ironic by clash + situational + other	gold-standard	C=2^3, γ=2^{-7}	67.76%
SVM+clash feat.	ironic by clash + situational + other	automatic	C=2^1, γ=2^{-7}	67.92%

Table 6.5: Optimisation scores of the SVM+clash system for irony detection on the training data.

Next, the optimised model was applied to the test set, the results of which are shown in Table 6.6. In grey we added the baseline, which is the best SVM score obtained through our combined feature group experiments (i.e. lexical + semantic + syntactic features, see Chapter 4).

system	**positive class**	**implicit sentiment**	**accuracy**	**precision**	**recall**	**F$_1$**
baseline SVM (lex+sem+synt)	ironic by clash + situational + other	-	69.21	68.92	**71.34%**	70.11%
SVM+clash feat.	ironic by clash + situational + other	gold-standard	**69.83%**	**70.25%**	70.10%	**70.18%**
SVM+clash feat.	ironic by clash + situational + other	automatic	69.21%	68.92%	**71.34%**	70.11%

Table 6.6: Performance of the SVM+clash system for irony detection using gold-standard and automatic implicit sentiment information.

As can be deduced from Table 6.6, adding a contrast feature based on automatically derived implicit sentiment does not enhance the classification performance of the original SVM (70.11%). The scores being equal to the baseline, the feature seems to add no information to the model. When using gold-standard implicit sentiment information, however, F$_1$-score is slightly better (70.18%). Although the overall system performance does not show a substantial improvement over the baseline, it is worth to note that precision increases by 1.3 point. Taking into account that the feature space is large (i.e. 36,175 features), this might indicate the feature's importance to the classifier.

Based on the raw results, we can conclude that adding a polarity contrast feature enhances the performance of our SVM classifier. In a next step, we investigated whether the results of the clash-based system (based on gold-standard implicit sentiment) are significantly better than that of the original SVM classifier. In the same way as explained in Chapter 4, ten thousand bootstrap samples (n=

958) with replacement were randomly drawn from the output of the two systems and evaluated by means of F_1-score. Subsequently, a paired samples t-test was applied to compare the mean scores and standard error over all sample scores for both systems, which showed a significant ($p < 0.05$) difference.

To continue the evaluation of our system, and similarly to Section 6.2, it makes sense to report recall obtained for the *ironic by clash* category, and instances from that category for which no hashtag is required to understand the irony.

system	**positive class**	**implicit sentiment**	**recall**
baseline			
SVM (lex+sem+synt)	ironic by clash	-	78.11%
SVM (lex+sem+synt)	ironic by clash (no hashtag required)	-	89.82%
SVM+clash feat.	ironic by clash	gold-standard	76.04%
SVM+clash feat.	ironic by clash (no hashtag required)	gold-standard	88.62%
SVM+clash feat.	ironic by clash	automatic	78.11%
SVM+clash feat.	ironic by clash (no hashtag required)	automatic	89.82%

Table 6.7: Performance of the SVM+clash system for irony detection on *ironic by clash* using gold-standard and automatic implicit sentiment information.

We can observe from the table that the scores obtained by SVM+clash using automatic implicit sentiment do not differ from the original SVM classifier (*supra*). In effect, as we explained earlier, adding a contrast feature based on automatically derived implicit sentiment does not affect the classification performance (see Table 6.6).

In sum, the experiments revealed that the original SVM is hard to outperform using a polarity contrast method. In effect, adding a polarity contrast feature only causes a slight improvement, as precision goes up by 1.3 points, but recall of the system decreases by 1.2 points. It is worth noting however, that the original SVM classifier exploits a rich (and optimised) feature set including lexical, word- and character-based *n*-grams, which both have shown to work very well for irony detection (e.g. Jasso López and Meza Ruiz 2016), and a number of syntactic and semantic features. Moreover, one should bear in mind that, with over 36,000 information sources, the feature space is very large. This might limit the effect of adding one single feature to the space. Consequently, to better assess the importance of the contrast feature for this task, we believe that an essential direction for future research involves feature selection or feature weighing techniques, or the design of a meta-learning approach.

6.3.2 A hybrid system

While Section 6.3.1 describes the inclusion of a polarity contrast feature to enhance irony detection, in the following paragraphs we implement a hybrid system for irony detection by combining the output of the polarity contrast-detection system with that of our SVM classifier (see Chapter 4). In short, the output of both classifiers (i.e. one prediction per tweet) is considered for the final irony prediction of a tweet. Two conditions are implemented in the system to define whether an instance is ironic: i) both systems agree that a tweet is ironic (*AND-combination*), and ii) one of the two systems predicts the tweet as ironic (*OR-combination*). Similarly to the previous sections, evaluation of the contrast-based system is done using gold-standard and automatically derived implicit sentiment information.

Table 6.8 presents the results of the combined classifier output. As the baseline, we added the SVM approach as described in Chapter 4, the results of which are in grey. Finally, Table 6.9 presents the results of the system on the *ironic by clash* category.

	system	**positive class**	**implicit sentiment**	**accuracy**	**precision**	**recall**	**F$_1$**
	baseline SVM (lex+sem+synt)	ironic by clash + situational + other	-	**69.21%**	68.92%	71.34%	**70.11**
1	AND-combination	ironic by clash + situational + other	gold-standard	63.78%	**73.96%**	43.92%	55.11%
2	OR-combination	ironic by clash + situational + other	gold-standard	62.42%	59.15%	**83.30%**	69.18%
3	AND-combination	ironic by clash + situational + other	automatic	58.98%	69.01%	34.43%	45.94%
4	OR-combination	ironic by clash + situational + other	automatic	62.11%	58.97%	82.68%	68.84%

Table 6.8: Performance of the hybrid approach to irony detection using automatic and gold-standard implicit sentiment information.

Table 6.8 reveals that, when looking at the combined setups, systems 1 and 2 outperform 3 and 4, as the former rely on gold-standard implicit sentiment information. Although the baseline scores best in terms of F$_1$-score, showing also a good balance between precision and recall, system 2 achieves a comparable result, but yields much higher recall. Logically, as shown by system 1, requiring both systems to predict an instance as ironic enhances precision of the system, but at the expense of recall.

When comparing the results to Table 6.6, we observe that, depending on how the two systems are combined (i.e. AND/OR), respectively precision and recall are better than including polarity contrast information as a feature for the SVM classifier. This demonstrates that polarity contrast information has a strong potential for improving irony detection, on the condition that the information provided by both systems is properly combined. The results in Table 6.8 also suggest that other methods besides a hybrid approach might enhance the performance of the SVM classifier based on polarity contrast information, for instance cascaded or ensemble-learning techniques.

	system	**positive class**	**implicit sentiment**	**recall**
	baseline			
	SVM (lex+sem+synt)	ironic by clash	-	78.11
	SVM (lex+sem+synt)	ironic by clash (no hashtag required)	-	89.82
5	AND-combination	ironic by clash	gold-standard	47.04
6	AND-combination	ironic by clash (no hashtag required)	gold-standard	75.45
7	OR-combination	ironic by clash	gold-standard	**87.57**
8	OR-combination	ironic by clash (no hashtag required)	gold-standard	**97.60**
9	AND-combination	ironic by clash	automatic	33.43
10	AND-combination	ironic by clash (no hashtag required)	automatic	47.90
11	OR-combination	ironic by clash	automatic	86.69
12	OR-combination	ironic by clash (no hashtag required)	automatic	95.81

Table 6.9: Performance of the hybrid approach to irony detection on *ironic by clash* using automatic and gold-standard implicit sentiment information.

Finally, Table 6.9 presents the results of the combined systems on the *ironic by clash* category. Recall in setups 7, 8 and 11, 12 being very high shows that i) when combining our original SVM with a polarity contrast based system (OR-combination), we were able to recognise almost all *ironic by clash* instances, and about 85% of the instances where in fact a hashtag is required.

Interestingly, while Tables 6.1 and 6.3 showed a clear drop in performance when automatic implicit sentiment information was used versus gold-standard, the effect wore off from the moment the clash-based system was combined with the original SVM classifier (see Tables 6.6, 6.7, 6.8 and 6.9), showing that the latter covered for mistakes made by the SVM+clash system. Recall for the AND-combinations (i.e. setups 5, 6, 9 and 10) being much lower underlines that the SVM and contrast-based systems for irony detection are complementary, each being able to capture specific realisations of *ironic by clash* instances. Most likely, as the SVM-based approach exploits lexical, syntactic and semantic features, it is able to recognise instances of irony where no polarity contrast could be detected thanks to other clues such as punctuation, interjections, Part-of-Speech tags, and so on.

The results further demonstrate that our approach compares favourably with Riloff et al. (2013). They make use of a bootstrapping approach to collect explicit positive phrases (e.g. 'I love') and negative verb phrases (e.g. 'being ignored') to model an [imp-exp] polarity contrast and report a recall of 44% when their SVM classifier (exploiting bag-of-words features) is combined with the contrast method they implemented.

6.4 Summary

In this chapter, we explored to what extent irony detection benefits from polarity contrast information, the potential of which has recently been suggested in irony literature. In concrete terms, we compared the performance of a state-of-the-art SVM classifier for irony (see Chapter 4) before and after it was informed with polarity contrast information.

As a first step, a polarity contrast-based irony detection system was developed. A key challenge in recognising polarity contrasts is recognising implicit sentiment or connotative knowledge (i.e. stereotypical sentiment related to particular situations). To tackle this problem, we made use of the Twitter method described in Chapter 5. Existing sentiment lexicons were used to identify explicit sentiment expressions. The evaluation of the system was done in different setups (e.g. by making use of gold-standard and automatically defined implicit sentiment) and on different types of irony (i.e. all types versus *ironic by clash*) and revealed that, although the contrast-based system does not outperform our original SVM classifier, it proves to be a strong indicator for irony, as it achieves high precision when a contrast between explicit and implicit sentiment is detected. Moreover, the system is able to identify a number of ironic instances that the SVM classifier overlooks. The system is prone to overgeneration, however, when also explicit polarity contrasts are taken into account, as these are likely to occur in non-ironic tweets as well.

Secondly, we combined the output of the contrast-based approach to irony detection with the SVM classifier by i) including the former as a feature in the SVM model and by ii) combining the output of the two systems into a hybrid system for irony detection. The experiments showed that including polarity contrast information (based on gold-standard implicit sentiment) as a feature yields a small improvement in precision over the original SVM classifier (i.e. 70.25% versus 68.92%), but recall drops slightly. Finally, combining the two classifiers into a hybrid system resulted in higher precision (*AND-combination*) and higher recall (*OR-combination*) than the SVM classifier, depending on the implementation. When evaluated on the *ironic by clash* category, the system achieved a recall of up to 88% and even 98% on instances of this class where no irony-related hashtag is required to notice the irony.

Although the original SVM classifier appears hard to beat when performing irony detection, a number of interesting observations were made throughout this chapter. To begin with, the experiments revealed that irony detection clearly benefits from polarity contrast information. Finding such polarity contrasts is, however, a challenging task due to several reasons. First, as we concluded in Chapter 5, defining implicit sentiment is challenging and studies are still scratching the surface of this task. We made use of crawled Twitter data to define implicit sentiment (or connotative knowledge) and concluded that finding (sufficient) tweets, especially for very specific situation phrases, constitutes the main challenge in this approach. Second, our experiments revealed that relying on a lexicon-based approach to find explicit sentiment expressions on Twitter also appeared challenging sometimes, as sentiment expressions often contain concatenated hashtags (e.g. '#ilovethis'), creative spelling (e.g. 'yaaaaayyy') and slang (e.g. 'swag'), and other expressions which could not be recognised using sentiment lexicons (e.g. '[...] is exactly what I need'). Third, our corpus analysis (see Chapter 3) revealed that a substantial part of ironic contrasts are realised through an irony-related hashtag (e.g. 'Bieber's concert was so awesome yesterday #not'), hence they are difficult if not impossible to identify without such hashtags. Fourth, detecting explicit sentiment contrasts as a clue for irony is prone to overgeneration, as non-ironic tweets may also contain words with contrasting polarities (e.g. when talking about the advantages and drawbacks of something, or discussing positive and negative aspects of an object). As such, optimising the sentiment lexicon-based approach or adopting supervised sentiment analysis to recognise explicit sentiment constitutes an important direction for future work.

In conclusion, exploiting implicit sentiment or connotative knowledge is a relatively new research direction in automatic irony detection. However, similar work has been done by (Riloff et al. 2013), who made use of bootstrapped learning to extract positive sentiment and negative situation phrases from hashtag-labelled ironic tweets (see Chapter 2). Their combined method (i.e. contrast-based system + SVM classifier) yielded an F-score of 51% and recall of 44%, so we can conclude that the results of our hybrid approach compare favourably to their approach. Moreover, while their method requires a large irony corpus to extract implicit sentiment phrases, we were able to recognise implicit sentiment based on real-time Twitter data, without requiring any training data. As opposed to the researchers, however, we did not address the problem of identifying text spans carrying implicit sentiment. Instead, we relied on manually annotated situation phrases and their related implicit sentiment (or connotation). Identifying such phrases automatically in tweets will, however, be an interesting direction for further research.

CHAPTER 7

Using irony detection for sentiment analysis: a use case

Irony detection has been stated to have a large potential for improving sentiment analysis (Barthi et al. 2016, Gupta and Yang 2017, Lunando and Purwarianti 2013, Maynard and Greenwood 2014). With the present use case, we aim to provide an answer to our third research question **"can our automatic irony detection approach enhance state-of-the-art sentiment classification?"** In Chapter 4, we developed an automatic system for irony detection exploiting lexical, semantic and (shallow) syntactic features, which was extended by a polarity contrast feature in Chapter 6. To provide an extrinsic evaluation of the system, this chapter explores to what extent sentiment analysis benefits from automatic irony detection. In concrete terms, we test the performance of an optimised sentiment classifier on a set of ironic tweets before and after the sentiment classifier is informed with irony information.

We start this chapter with a detailed description of our sentiment analysis pipeline, and then present the results of a hybrid system that combines sentiment and irony information to provide a final sentiment label for a set of tweets.

7.1 Automatic sentiment analysis

The following paragraphs provide a brief introduction into the active research domain of sentiment analysis, after which we detail the development of our optimised sentiment pipeline.

7.1.1 Background

Finding its origin in the early 2000's, sentiment analysis has rapidly evolved towards one of the most dynamic research areas in natural language processing (Liu 2012). It is not by chance that its expansion coincided with the growth of social media, providing the machine learning community with an unseen amount of subjective, user-generated content. Sentiment analysis is concerned with modelling subjective information (i.e. sentiments, attitudes, opinions) in online text. Due to its importance in both research (e.g. natural language processing, sociology) and industry (i.e. large companies, as well as specialised startups), studies in the domain are numerous and significant progress has been made in recent years.

Among the most visible results of this active research area are the specialised shared tasks organised in the framework of SemEval (Workshop on Semantic Evaluation)[1], an international ongoing series of evaluations of semantic analysis systems. Within the framework of such tasks, participating teams can develop and submit a computational system based on data provided by the organisers. All teams making use of the same dataset and being evaluated equally allows to compare the performance of participating systems. As such, a lot of benchmark datasets and results have recently been provided to the research community.

Although continuous progress is being made in the field, an important bottleneck of sentiment analysis remains the frequent use of irony in social media data. The SemEval-2014 shared task *Sentiment Analysis in Twitter* (Rosenthal et al. 2014) demonstrated the impact of irony on automatic sentiment analysis by including a set of ironic tweets as an additional test set for the participating systems. The task results showed that, while sentiment classification performance on regular tweets reached up to F_1= 70.96%, scores on the irony test set varied between 28.96% and 56.50%. This considerable drop in performance demonstrates that sentiment classifiers require modifications when applied to ironic text.

We already raised the challenge of irony in sentiment analysis in Chapter 1 by presenting the following examples:

[1] http://alt.qcri.org/semeval2018

(70) I love how my mom says she can count on Rion more than me. #not #jealous.

(71) I feel so blessed to get ocular migraines.

(72) Go ahead drop me hate, I'm looking forward to it.

The benefits of irony detection for sentiment analysis on Twitter have already been explored in previous work (e.g. Barthi et al. 2016, Gupta and Yang 2017, Lunando and Purwarianti 2013, Maynard and Greenwood 2014). However, while most of these studies take a rather basic approach (i.e. using sentiment lexicons or *n*-grams) to sentiment and irony classification, the contribution of the present research is the combination of a complex feature-based irony detection system with an optimised sentiment classifier. Moreover, we present an extensive qualitative analysis that provides insights into the system performance on the different sentiment classes (i.e. positive, negative, neutral), and on the different types of irony (see Chapter 3).

7.1.2 System description

The system we present has been developed in the framework of the SemEval-2014 task on *Sentiment Analysis in Twitter* where it ranked sixteenth among fifty submissions (Van Hee et al. 2014). In a series of follow-up experiments, we optimised the model by means of feature selection and optimisation of the algorithm's hyperparameters, the results of which are presented in Section 7.1.3.

It is worth noting that the sentiment classifier was already briefly introduced in Chapter 5, where it was used to automatically define the prevailing sentiment about a specific phrase or situation in a set of crawled tweets. However, no detailed description nor evaluation of the pipeline has been presented yet.

Dataset and preprocessing

The train and test corpus for the system were distributed in the framework of the shared task. While the training corpus contains merely Twitter data, the test corpus is composed by a variety of user-generated content, including tweets (i.e. regular and ironic), text messages (SMS), and blog posts (i.e. retrieved from LiveJournal). Table 7.1 presents the corpus statistics.

Both the training and test corpus contain three class labels expressing the sentiment of each instance, namely positive, negative, and neutral, which represent

	training corpus	**held-out test corpus**			
	Twitter	Twitter (reg.)	Twitter (sarc.)	SMS	blog
	11,338	5,666	86	2,093	1,142
total	**11,338**	**8,987**			

Table 7.1: Distribution of the training and test corpus of the sentiment classifier.

37%, 16% and 47% of the dataset, respectively. Below are presented some corpus examples and the corresponding sentiment label.

(73) @username nice piece if exciting news that may make you happy... Wizards of waverly place is coming back! **(positive)**

(74) @username lmao i sat here for five minutes like what the fuck did i do to courtney???? Ha damn........ -_- **(negative)**

(75) Yearbook pictures for Jr.Larc's will be next Tuesday, the 6th at lunch at the library parking lot side **(neutral)**

As explained in Chapter 4, a number of preprocessing steps have to be taken prior to feature extraction based on the experimental corpus. To recapitulate, preprocessing refers to all steps that are needed for formatting and cleaning the collected tweets and enriching the data with the linguistic information required for feature engineering. With the exception of dependency parsing, the same preprocessing steps were undertaken as described in Chapter 4. We therefore explain the dependency parsing step here, and refer to Chapter 4 for more details about the other preprocessing steps.

The following preprocessing steps were taken:

- Data cleaning
- Tokenisation
- Part-of-Speech tagging
- Named entity recognition (NER)
- **Dependency parsing:** linguistic analysis process that identifies grammatical relations between words in a sentence. We made use of the caseless parsing model of the Stanford parser (de Marneffe et al. 2006). Dependency relations are represented by the name of the relation and the relation's governor and dependent. In what follows, we show the output of the dependency parser for example sentence 76.

(76) My dog also likes eating sausage[2].
Typed dependencies: poss (dog-2, My-1), nsubj (likes-4, dog- 2), advmod (likes-4, also-3), root (ROOT-0, likes-4), xcomp (likes-4, eating-5), dobj (eating-5, sausage-6) .

- (Shallow) normalisation

Information sources

Before creating the sentiment model, a number of information sources (i.e. *features*) were extracted to provide the model with relevant information for the task:

- **Bag-of-words features (BoW):** token unigrams, bigrams and trigrams, as well as character trigrams and fourgrams (without crossing token boundaries). *N*-grams that occurred only once in the training corpus were discarded to reduce feature sparseness.
- **Post length:** numeric feature indicating the number of tokens contained in each tweet.
- **Word-shape features:** set of numeric and binary features including character and punctuation flooding, punctuation tokens, the number of upper cased tokens in the tweet, and the number of hashtags in the tweet.
- **Part-of-Speech (PoS) features:** four features for each one of the 25 tags in the PoS-tagset, indicating i) whether the tag occurs in the tweet or not, ii) whether the tag occurs zero, one, or two or more times, iii) the absolute and iv) relative frequency of the tag.
- **Dependency relation features:** four binary features for every dependency relation found in the training data (cf. example 77). The first feature indicates the presence of the lexicalised dependency relations in the test data (hm-lex). For the remaining features, the dependency relation features are generalised in three ways, as proposed by Joshi and Penstein-Rosé (2009): by backing off the head word to its pos-tag (h-bo), the modifier word to its pos-tag (m-bo), and both the head and modifier word (hm-bo).

  (77) I had such a great time tonight that I've decided to keep celebrating!
  → hm-lex: amod (time, great), h-bo: amod (N, great), m-bo: amod (time, A), hm-bo: amod (N, A)

[2] Example taken from http://nlp.stanford.edu.

- **Named entity features:** four features indicating the presence of named entities in a tweet: one binary feature (the tweet contains a NE or not) and three numeric features, indicating i) the number of NEs in the tweet, ii) the number of tokens that are part of a NE, and iii) the frequency of NE tokens in the tweet.

- **PMI features:** two numeric features based on PMI (*pointwise mutual information*) obtained from i) word-sentiment associations found in the training data, and ii) an existing PMI lexicon (Mohammad and Turney 2013). A positive PMI value indicates positive sentiment whereas a negative score indicates negative sentiment. The higher the absolute value, the stronger the degree of association with the sentiment. PMI values were calculated by subtracting a word's association score with a negative sentiment from the word's association score with a positive sentiment, as shown in the following equation:

$$PMI(w) = PMI(w, positive) - PMI(w, negative) \tag{7.1}$$

- **Sentiment lexicon features:** four sentiment lexicon features (i.e. the number of positive, negative and neutral words, and the overall tweet polarity) based on existing resources: AFINN (Nielsen 2011), General Inquirer (GI) (Stone et al. 1966), MPQA (Wilson et al. 2005), the NRC Emotion Lexicon (Mohammad and Turney 2013), Liu's opinion lexicon (Liu et al. 2005), Bounce (Kökciyan et al. 2013), and our own emoticon lexicon derived from the training corpus. The sentiment lexicon features were extracted by looking at all the tokens in the instance and hashtag tokens only (e.g. *win* from *#win*). Negation cues were taken into account by flipping the polarity of a sentiment word if it occurred within a window of three tokens at the left or right of a negation word (e.g. 'never', 'not'). It is noteworthy that we took a window of only one word for negation cues in Chapter 4, since we noticed that the irony dataset, as compared to the sentiment tweets that were distributed in 2014, are more fragmented (i.e. by hashtags, punctuation, emoji, etc.).

Comparison with other participating teams reveals that similar features were exploited by the top-performing systems for the task (e.g. Günther et al. 2014, Miura et al. 2014, Tang et al. 2014), except that some of them also included word-embeddings and cluster features.

7.1.3 Experimental setup and model optimisation

The main objective was to build a sentiment classifier that makes optimal use of the information explained in the above section. We made use of support vector machines (as implemented in the LIBSVM library (Chang and Lin 2011)) as they have shown to outperform other classifiers for this task (e.g. Al-Mannai et al. 2014, Bin Wasi et al. 2014, Zhu et al. 2014). Following the best practices as described in Hsu et al. (2003), all datasets for the experiments were scaled to the range [0, 1] before building models using LIBSVM to avoid large feature values being given more weight in the model than smaller values.

Given that SVMs can take a varied set of hyperparameter values (Chang and Lin 2011), and that we chose to extract a rich set of features that are potentially informative for this task, we chose to optimised our model by means of hyperparameter tuning and feature (group) selection. To this end, we made use of the Gallop toolbox (*Genetic Algorithms for Linguistic Learner Optimisation*) Desmet et al. (2013). The algorithm allows for *joint optimisation*, meaning that feature selection and hyperparameter optimisation are performed simultaneously so that heir mutual influence can be evaluated. This way, optimal hyperparameter settings could be defined for the algorithm and the most informative features were selected.

To reduce experimental complexity when using Gallop, the original feature space ($>$ 400,000 features) was first filtered using information gain (IG) (Daelemans et al. 2009), which measures the difference in entropy (i.e. the uncertainty about a class label given a set of variables) when the feature is present or absent in a feature vector representation. All features with an information gain below 0.001 were discarded, which resulted in a new feature space of 1,850 features. The threshold was empirically defined so as to keep a (rather) limited set of features. After feature filtering using IG, the remaining features were grouped into 36 feature groups for the wrapped feature selection with Gallop (Desmet et al. 2013). Out of the variety of hyperparameters that can be set for a LIBSVM classifier, we chose to optimise kernel-specific settings including t (the kernel type), d (the kernel function degree), γ (the kernel function gamma), and the classification cost value C. We do not go into the details of the optimisation process here, but refer to (Van Hee et al. Submitted) for a comprehensive overview of the optimisation experiments and results.

Optimisation of the classifier was done by means of cross-validation on the training data. During this optimisation process, both the feature groups and hyperparameters were defined so as to maximise macro-averaged F_1-score. As explained earlier (see Section 4.4.2), when performing multiclass classification, one is generally interested in the system performance of different class labels, as opposed to binary classification or detection tasks with only one label of interest. To have an idea of the general performance of our sentiment classifier, F_1-scores for the different class labels are averaged. There exist two methods for doing this: i) by macro-averaging, where metrics are calculated for each label, after which an unweighted average is computed and ii) by micro-averaging, which calculates metrics globally by counting the total true positives, false negatives and false positives over all class labels (Manning et al. 2008). Macro-averaging was preferred to micro-averaging to avoid the system being biased towards the majority sentiment class (Sokolova and Lapalme 2009).

7.1.4 Results

This section presents the results of the classification experiments. Accuracy and (macro-averaged) precision, recall and F_1-score are reported as the evaluation metrics. Table 7.2 shows the cross-validated results of our sentiment classifier in three steps of the experimental setup. The first and second setup describe the classifier being applied in its default hyperparameter settings and exploiting the full and IG-filtered feature set, respectively. In the third setup, we describe the results of the joint optimisation experiment (i.e. hyperparameter optimisation and wrapped feature (group) selection), after the feature groups had been filtered using information gain.

	setup	**accuracy**	**precision**	**recall**	**F_1**
1	full feature set	46.97%	15.66%	33.33%	21.30%
2	filtered feature set (IG)	67.25%	76.90%	52.11%	48.67%
3	filtered (IG + Gallop)	**79.63%**	**77.64%**	**74.68%**	**75.84%**

Table 7.2: Cross-validated results obtained with the full and filtered feature sets applying the sentiment classifier in its default hyperparameter settings (1+2) and after joint optimisation (3).

As can be deduced from the table, applying feature selection results beneficial to the classification performance, as it decreases sparseness and removes redundant information. Analysis of the optimisation experiment revealed that the linear kernel (t= 0) was always selected, and that the optimal cost value (C) for the five best individuals was 0.25. No optimal values were defined for d and γ as the parameters are irrelevant when using a linear kernel. The analysis further revealed that important features for the task include bags-of-words, *flooding*, named entity features, sentiment lexicon features, and PMI features.

Having in place an optimised sentiment model, in a next step we tested the Twitter model on the held-out corpus containing a variety of genres, the results of which are presented in Table 7.3. It should be noted, however, that we report scores obtained by evaluating the systems on the three classes, being positive, negative and neutral. For the competition, evaluation was based only on the positive and negative class.

SMS2013	**TWE2013**	**TWE2014**	**TWE2014Sarcasm**		**LiveJour.2014**	**full test**
2.093 inst.	3.813 inst.	1.853 inst.	86 inst. (ori)	76 inst. (corrected)	1.142 inst.	8.987 inst.
70.53%	66.36%	64.83%	40.90%	16.58%	68.00%	67.28%

Table 7.3: Sentiment classification performance (macro-averaged F_1-score) on different genres in the SemEval-2014 test set.

As can be deduced from this table, the model performs well on different social media genres, even genres that were not included in the training set (i.e. SMS, blog), which indicates its robustness to data genres other than Twitter. When comparing the results for the full test set, we observe that our system now compares favourably to the winning system by Miura et al. (2014), which obtained an F_1-score of 65.40%. This clearly shows that optimisation of the classifier by means of feature selection and hyperparameter tuning pays off.

Finally, it is worth to note that we report two scores for the *TWE2014Sarcasm* set, namely *ori* and *corrected*. While *ori* presents the result of our system obtained on the original *TWE2014Sarcasm* set, *corrected* presents the system performance after the gold-standard labels of the test set were corrected. In fact, a qualitative analysis of the system output revealed that the labels showed a number of inconsistencies that needed correction. We therefore decided to re-annotate the *Twitter2014Sarcasm* set according to our irony guidelines (see Chapter 3). As such, approximately half of the instances received a new gold-standard sentiment label and ten instances were removed as we considered them non-ironic. As a result of this correction, the scores of our system dropped considerably. When discussing the results of our irony-sensitive sentiment analyser in the next section, we will compare the results to the *corrected* score.

7.2 Irony-sensitive sentiment analysis

As our optimised sentiment classifier struggles to define the correct sentiment in ironic tweets, we explored the potential of automatic irony detection for this task. The following paragraphs describe the development of a hybrid system combining the sentiment classifier with the irony detection system (i.e. SVM+clash) we described in Chapter 6.

7.2.1 Dataset

To test our hypothesis, we made use of the ironic instances as part of the SemEval-2014 test data (*TWE2014Sarcasm*). The corpus is named 'TWE2014-Sarcasm', but like in the remainder of this thesis, we will refer to the instances as 'ironic'. As we mentioned earlier in this chapter, we corrected a number of instances in the dataset that we considered incorrect, as exemplified in sentences 78 and 79.

(78) And the boss just posted the schedule, and I work this saturday. Yay OT. **(positive)**

(79) More snow tomorrow... fantastic. #ihatesnow **(positive)**

The examples show two ironic tweets whose sentiment label we modified to *negative*, since both clearly use irony to express a negative sentiment.

Given that the dataset only contains ironic tweets, we created a second test set to properly evaluate the performance of our irony classifier in a balanced distribution. For this purpose, we expanded *Twitter2014Sarcasm* with 76 instances from *TWE2014*, the regular (i.e. non-ironic) Twitter test set that was provided for the shared task. Hence, we will report results of our system on two corpora, namely *Twitter2014Sarcasm* and *Twitter2014Sarcasm-balanced*.

Table 7.4 provides insights into the class distribution in both datasets, showing that the majority of the tweets carry a negative sentiment, which is also shown in our irony corpus (see Chapter 3).

dataset	**positive class**	**negative class**	**neutral class**
Twitter2014Sarcasm	1%	91%	8%
Twitter2014Sarcasm-balanced	23%	53%	24%

Table 7.4: Class distribution in *Twitter2014Sarcasm* and *Twitter2014Sarcasm-balanced*.

7.2.2 Experimental setup

To predict irony in both datasets, we made use of the SVM+clash system as described in Chapter 6. To recapitulate, the system exploits lexical, semantic and syntactic features, as well as a polarity contrast feature. To define this polarity contrast in the current experiments, we made use of gold-standard implicit sentiment information to limit errors percolating from this step. In a second step, irony detection was applied to the two corpora, and the irony predictions were subsequently used to inform the sentiment classifier. In concrete terms, a post-processing implied that the predicted sentiment for a particular instance was inverted if it had been predicted as ironic. As such, a positive sentiment label became a negative one, and vice versa. Neutral instances remained neutral if irony was detected.

To evaluate the performance of this hybrid system, we report accuracy, precision, recall and F$_1$-score. While in the irony detection experiments (see Chapters 4 and 6), the latter three metrics were calculated on the positive class instances only, they are macro-averaged over the different class labels in the present chapter. We refer to Section 7.1.3 for more details about macro-averaged performance metrics. We also report accuracy, which equals micro-averaged scores (Sokolova and Lapalme 2009) in this case, as it weighs class labels proportionally to their frequency and therefore favours larger classes. As explained in Chapter 4, on the one hand, accuracy may be misleading when the distribution is unbalanced, as frequent class labels get more weight in the global score. On the other hand, the strong effect of minority classes on macro-averaged F$_1$-score makes that it does not fully reflect the overall performance on a typical (skewed) distribution. Taking into account the unbalanced (sentiment) class distribution in the *Twitter2014Sarcasm* datasets (see Table 7.4), we choose to report both measures.

7.2.3 Results

Tables 7.5 and 7.6 present the results of our sentiment classifier respectively before and after irony information was taken into account for the final sentiment prediction.

dataset	**accuracy**	**precision**	**recall**	**F$_1$**
Twitter2014Sarcasm	17.11%	30.76%	54.35%	16.58%
Twitter2014Sarcasm-balanced	42.11%	49.57%	51.32%	41.80%

Table 7.5: Performance of the sentiment classifier without taking irony information into account.

Table 7.5 reveals that the performance without irony information is low. As discussed earlier in this section (see Table 7.3), we report scores on a corrected version of the *Twitter2014Sarcasm* dataset, because part of the original data were labelled incorrectly and this inflated classification performance. The low performance obtained after correcting the gold labels confirms that sentiment analysis is strongly affected by irony presence (e.g. Liu 2012, Maynard and Greenwood 2014). Unsurprisingly, performance on the balanced corpus is better, since the extra difficulty that irony presents to a sentiment classifier is present in only half of the instances. As can be noticed, F_1-scores are clearly lower than precision and recall. This can be explained by the fact that precision and recall are largely unbalanced for the different class labels. More precisely, we observe that precision is very low for the positive and neutral class because the sentiment classifier tends to overgenerate both classes due to their low frequency in the dataset (see Table 7.4). By contrast, recall for both classes is high. The opposite is observed for the negative class, where recall is low (i.e. many of the ironic instances contain overtly positive language and are consequently predicted as positive), and precision high. As such, the severe imbalance between precision and recall results in low F_1-scores for all classes. However, macro-averaging precision and recall results in less dramatic scores, because low recall for one class (e.g. negative) is compensated by high recall for another (e.g. positive).

To provide the sentiment classifier with irony information, we made use of the SVM+clash system described in Chapter 6 to predict ironic instances in both datasets. Having at hand a sentiment and irony prediction for each instance, we subsequently implemented a hybrid system that combined both pieces of information: if irony had been detected in an instance, the sentiment prediction for that instance was inverted (except for the *neutral* class).

dataset	**accuracy**	**precision**	**recall**	**F_1**
Twitter2014Sarcasm	59.21%	40.56%	69.81%	36.71%
Twitter2014Sarcasm-balanced	53.95%	50.32%	53.46%	49.23%

Table 7.6: Performance of the sentiment classifier with automatically derived irony information.

Table 7.6 presents the results of this hybrid sentiment classifier and clearly shows that incorporating irony information enhances sentiment classification performance. In effect, F_1-score rose by no less than 20% for the *Twitter2014-Sarcasm* set, and by 7% for the balanced set. Performance increases for accuracy were even more outspoken, with 42 and 11 points, respectively. These results clearly demonstrate the usefulness of automatic irony detection for sentiment classification.

F_1-scores are considerably lower than accuracy. As discussed in Section 4.4.2, this can be explained by the behaviour of macro-averaged F_1-score, which weighs each class equally, regardless of the class distribution. To understand this better, we present the scores per class in Table 7.7, which reveals that performance on the positive class, which is strongly underrepresented in *Twitter2014Sarcasm*, is considerably lower than the performance on the negative and neutral classes.

With macro-averaging, the overall score is affected much more by the low performance on positive instances than accuracy is, as the latter weighs each class proportional to its distribution.

dataset	positive class	negative class	neutral class
Twitter2014Sarcasm	12.50%	74.55%	23.08%
Twitter2014Sarcasm-balanced	25.40%	63.24%	59.05%

Table 7.7: Performance (macro-averaged F_1-score) of the sentiment classifier with irony information on the different sentiment classes.

The results in Table 7.6 demonstrate the benefits of irony detection for automatic sentiment analysis, but still show a dip in sentiment analysis performance on ironic datasets as compared to regular, non-ironic datasets (cf. Table 7.3). This can be explained by two observations. Firstly, it appears that the sentiment classifier suffers from the skewed class distribution in both datasets. As shown in Table 7.4, both datasets are heavily biased towards the negative class. As the classifier is trained on a differently distributed and more balanced Twitter corpus, it tends to overgenerate positive and neutral class instances, which negatively affects precision for both classes. Moreover, given that ironic tweets often contain strongly positive language, negative instances were often predicted as positive, which negatively affected recall for the negative class. Secondly, part of the classification mistakes can be explained by errors percolating from the irony detection step. In *Twitter2014Sarcasm* and *Twitter2014Sarcasm-balanced* respectively 29% and 50% of the classification errors percolate from an erroneous irony prediction.

To get a sense of the potential sentiment classification performance improvement with 100% accurate irony detection, we show the results of our classifier with access to gold-standard irony information in Table 7.8.

dataset	accuracy	precision	recall	F_1
Twitter2014Sarcasm	60.53%	37.58%	37.44%	33.06%
Twitter2014Sarcasm-balanced	63.82%	64.31%	64.54%	61.35%

Table 7.8: Performance of the sentiment classifier with gold-standard irony information.

As shown in the table, accuracy goes up by 1.3 point on *Twitter2014Sarcasm* and by 10 points on the balanced dataset, as compared to Table 7.6. In other words, the performance with automatic irony prediction comes quite close to this performance ceiling. Surprisingly, F_1-score on *Twitter2014Sarcasm* is lower when gold-standard irony information is used to inform the sentiment classifier. This can be explained by 12 negative instances that are wrongly predicted as positive, which affects precision for the positive class and recall for the negative class. For these tweets (e.g. example 80), the original sentiment prediction was correct, but an erroneous irony prediction for these instances caused the final sentiment label to be inverted.

(80) Spring pictures tomorrow = drama drama drama.... Can't wait til morning.

Most likely, the error occurs because the explicitly negative words in the target ('drama drama drama') influence the sentiment classifier more than the seemingly positive evaluation ("can't wait til morning") that is inverted by irony. In order to avoid such errors, irony detection and sentiment classification systems would have to be more tightly integrated and take the scope of irony into account.

Finally, like in the previous chapters, we looked into the system performance for the specific subcategories of irony, namely *ironic by clash* and *other irony* (no realisations of situational irony were present in the *Twitter2014Sarcasm* test set). Not surprisingly, analysis revealed that when looking at the category *ironic by clash*, the sentiment classifier benefits much more from irony detection (i.e. accuracy + 50%) than for the category *other irony* (i.e. accuracy + 11%). In effect, while inverting the literal polarity is essential to understand the intended message in the former category, it is not for the category *other* irony. For an irony-sensitive sentiment classifier to perform well on this category, fine-grained irony detection would be required, allowing to define the type of irony and invert the original sentiment of an instance depending on the type of irony that is detected.

7.3 Summary

With the present use case, our goal was to conduct an extrinsic evaluation of our irony detection system. As sentiment classifiers have shown to struggle with ironic text (e.g. Barthi et al. 2016, Gupta and Yang 2017, Lunando and Purwarianti 2013, Maynard and Greenwood 2014), we explored to what extent our sentiment classifier benefits from automatic irony detection. To this end, we developed a hybrid system that takes irony information into account when defining the sentiment label for an instance. The system makes use of our automatic irony detection system (SVM+clash) as described in Chapter 6 and uses its output to inform the optimised sentiment classifier.

This classifier makes use of a support vector machine (SVM) and is trained on

English tweets provided in the framework of the SemEval-2014 task *Sentiment Analysis in Twitter* (Rosenthal et al. 2014). By making use of a varied set of features, the system ranked sixteenth among fifty submissions. Follow-up experiments involved optimisation of the classifier by means of feature filtering and hyperparameter optimisation using the genetic algorithm Gallop (Desmet et al. 2013). We observed that the results of the optimised classifier outperformed the winning system of the shared task, which demonstrates that optimisation pays off.

Having in place an optimised entiment classifier, we applied it to a test sets containing ironic tweets (Rosenthal et al. 2014) in a balanced and unbalanced distribution and inverted the predicted sentiment label of an instance if irony had been detected. The results of these experiments revealed that sentiment classification clearly benefits from automatic irony detection, showing a performance increase of 20% to 40% (F_1-score versus accuracy) on the *SemEval2014Sarcasm* (i.e. unbalanced) corpus, and of 7% to 12% on the *SemEval2014Sarcasm-balanced* set. A qualitative analysis revealed that classification errors are due to i) the skewed class imbalance in both corpora, which mainly affects the classifier's performance on the positive and negative class, and ii) errors percolating from the irony detection step. Indeed, testing the classification performance with perfect (i.e. gold standard) irony detection revealed that the performance with automatic irony prediction comes relatively close to this performance ceiling, and that the performance of the sentiment classifier is comparable to performance on regular (i.e. non-ironic) data (cf. *TWE2014* in Table 7.3).

Finally, the analysis revealed that the system performs best on *ironic by clash* instances, as other instances of irony do not necessarily require the explicit polarity being inverted (e.g. "Who wants to work for me tomorrow? Don't all stand up at once now"). To tackle this problem, fine-grained irony detection is required to inform the sentiment classifier about the type of irony being used (i.e. *ironic by clash*, or *situational irony* or *other irony*). This way, whether the sentiment prediction should be inverted or not would depend on the type of irony that is recognised. Another important direction for future work involves optimisation of the irony detection system, as similar experiments for the sentiment classifier have shown to pay off (Van Hee et al. Submitted).

CHAPTER 8

Conclusion

This thesis set out to explore automatic irony detection on social media, a topic that has attracted much research interest recently. Irony presents an important bottleneck to text mining systems that are traditionally trained on regular (i.e. non-ironic) data, one of the best known probably being sentiment analysis (Liu 2012). Although various systems and resources have been developed for different languages in the past few years, most studies have focussed on irony detection itself by using lexical clues, while studies on the mechanisms that underlie irony are scarcer (e.g. Karoui et al. 2015, Stranisci et al. 2016). Moreover, while irony detection is considered crucial to enhance sentiment analysis (e.g. Joshi, Bhattacharyya and Carman 2016, Maynard and Greenwood 2014), its actual effect on state-of-the-art sentiment classification has not sufficiently been investigated.

The main contribution of the current thesis is therefore a comprehensive approach to irony detection, starting with a manual annotation providing insights into the realisation of the phenomenon, varied sets of experiments for automatic irony detection, and finally an extrinsic evaluation of the system by means of a sentiment analysis use case. Moreover, as manually annotated corpora for irony detection are scarce, the dataset that has been developed in the current thesis will certainly be useful for further research. Another important contribution of this thesis are the exploratory experiments to automatically detect the implicit or prototypical sentiment related to particular situations. To this end, we compared a knowledge based and data-driven approach using respectively SenticNet 4 and Twitter. While SenticNet is a well-known lexico-semantics knowledge base, using real-time Twitter information to gain insights into the prototypical sentiment of particular phrases and situations has, to our knowledge, not been explored before.

We started this thesis by stipulating three main research questions for which a number of research objectives were defined (see Section 1.2). A series of experiments and analyses were conducted throughout this thesis to provide an answer to our research questions. In the following sections, we summarise our experimental findings, after which we discuss the limitations of this research and suggest some directions for future work.

8.1 Annotating irony in social media text

→ *Research question 1a: how is irony realised in social media text?*

To answer this research question, we collected and annotated a set of 3,000 English tweets using irony-related hashtags (i.e. *#irony, #sarcasm, #not*). A new annotation scheme was developed for this purpose, which is grounded in irony literature and allows for a fine-grained annotation. The scheme indicates different types of irony (i.e. *ironic by clash*, *situational irony* and *other irony*) and specific text spans within a tweet that realise the irony. We also provided the following working definition of irony; *"[verbal irony is] an evaluative expression whose polarity (i.e. positive, negative) is inverted between the literal and the intended evaluation, resulting in an incongruence between the literal evaluation and its context"*.

While related work often relies on hashtag labels to collect irony data, the annotation of our corpus revealed that 20% of the tweets containing an irony-hashtag were not ironic, which confirms our claim that manual annotations are critical to the current task. The fine-grained annotation scheme distinguishes instances of irony by means of a polarity clash, situational irony, and other irony. We observed that the majority of ironic tweets in our corpus (i.e. 72%) were realised by means of a polarity contrast, while situational and other irony account for 17% and 11% of the ironic instances, respectively. This observation confirms that irony generally involves saying the contrary of what is meant (e.g. Attardo 2000, Giora et al. 2005, Grice 1975).

No distinction between irony and sarcasm is made in this thesis, but it was observed that tweets that were considered harsh (i.e. carrying a mocking or criticising tone) more frequently contained the hashtag #sarcasm than #irony or #not. This would suggest that, when targeting entities, the phenomenon is more likely to be defined as 'sarcasm'. However, we do not consider this sufficient evidence that the two terms refer to distinct phenomena, and further research (e.g. using a larger corpus, multilingual data, etc.) would be necessary to confirm our findings.

Annotations below the tweet level revealed that in about half of the *ironic by clash* instances (i.e. 47%), an irony-related hashtag was required to perceive a polarity contrast. In the other half of these tweets, the polarity contrast mostly involved an explicit and implicit sentiment (so-called *target*) (e.g. 'Not being able to sleep is just excellent').

8.2 Detecting irony in social media text

> → *Research question 1b: can ironic instances be automatically detected in English tweets? If so, which information sources contribute most to classification performance?*

To answer the second part of our first research question, a series of binary classification experiments were carried out to detect irony automatically. In Chapter 4, we developed an irony detection system based on support vector machines (SVM). The algorithm was applied in its default kernel configuration, but optimal C and γ values were defined by means of a grid search in each experimental setup. As the experimental corpus, we made use of the manually annotated irony dataset described in Chapter 3, which was extended with non-ironic tweets from a background corpus until a balanced class distribution had been obtained. The final corpus (i.e. 4,792 tweets) was then divided into a set for training and a held-out test set to evaluate classification performance.

Irony detection using a rich feature set

We explored the potential of lexical, (shallow) syntactic, sentiment and semantic features as individual feature groups and combined. While similar features are commonly used in the state of the art, we expanded our lexical and semantic feature sets with respectively n-gram probabilities and word cluster information, two features that have, to our knowledge, not been sufficiently explored for this task. To this end, n-gram probabilities were derived from KENLM language models trained on an ironic and non-ironic background corpus. To provide the model with distributional semantics information, word embedding clusters were created using the Word2Vec algorithm.

Using the above-mentioned feature groups, a series of binary classification experiments were carried out and evaluated against three baselines: random class, word n-grams and character n-grams. The latter two proved to be very strong baselines, yielding an F_1-score of 66.74% and 68.40%, respectively, an observation that is in line with that of Buschmeier et al. (2014) and Riloff et al. (2013). The results showed that only the *lexical* feature group ($F_1 =$ 67.01%) outperformed the word n-gram baseline, but not the character baseline. An explanation for this could be that the former is mainly composed of bag-of-word features carrying information that is directly derived from the training data, while the other feature groups rely on external information (e.g. sentiment lexicons, background corpora). Nevertheless, all feature groups performed relatively well and therefore showed potential for irony detection.

Combining the feature groups showed that they are likely to provide complementary information, as combining the lexical with the syntactic and semantic feature groups caused a (statistically) significant performance increase. Yielding an F_1-score of 70.11%, this combined feature set outperformed the strong character baseline as well. In a final experimental round, classifiers of the individual feature groups were combined into a hybrid system (i.e. the output of one classifier is used to inform another) which benefitted recall, but at the expense of precision.

To get a better understanding of the bottlenecks in irony detection, a qualitative analysis of the classification output was done for the different types of irony. This showed that the classifier (exploiting lexical, semantic and syntactic features) performs best on *ironic by clash* and *situational irony* instances. It should not surprise, however, that performance is much lower on instances of the *other type of irony*, as the category assimilates realisations of irony that show neither a polarity contrast, nor situational irony, or that are ambiguous due to the lack of conversational context.

In sum, we developed a binary irony detection system using support vector machines and exploiting lexical, semantic and syntactic information sources. We found that the system performs best on ironic instances showing a polarity contrast, although the category presents some challenges as well (see further). Yielding an F_1-score of 70.11%, the system compares favourably to the work by González-Ibáñez et al. (2011) and Riloff et al. (2013).

Irony detection using polarity contrast information

In Chapter 6, we investigated to what extent irony detection benefits from polarity contrast information, the potential of which has often been underlined in irony literature, including the present thesis. Concretely, we explored the performance of a classifier exploiting merely polarity contrast information, and the added value of this information for our original SVM classifier.

A key challenge in recognising polarity contrasts is detecting implicit sentiment, for which we made use of the Twitter method described in Chapter 5 (see further). To define explicit sentiment, we made use of existing sentiment lexicons for English. The polarity contrast method for irony detection was implemented in two ways, i.e. by making use of gold-standard and automatically detected implicit sentiment.

The experiments revealed that, although the contrast-based system does not outperform our original SVM classifier, polarity contrast information appears to be a strong indicator for irony as it yields high precision when a contrast between explicit and implicit sentiment is found. Moreover, the system is able to identify a number of ironic instances that the SVM classifier overlooks.

We combined the output of the contrast-based system with that of the SVM i) by means of a polarity contrast feature and ii) in a hybrid system for irony detection. The results revealed that the contrast feature yields a small improvement in precision (i.e. 70.25% versus 68.92%), while recall shows a minor drop. Combining the classifiers into a hybrid system resulted in higher precision (AND-combination) and higher recall (OR-combination) than the original SVM classifier. Interestingly, when evaluated on the *ironic by clash* category, the hybrid system achieves a recall of 88%. On *ironic by clash* instances where no hashtag is required to sense the irony, recall even reached 98%.

Limitations and future work

While our experiments describe manual feature group selection, applying individual feature selection will be a crucial direction for future work to i) optimise the classifier by removing redundant information and ii) gain more insights into the most contributive features for this task.

Analysis of our experimental results revealed that the SVM system performs best on ironic instances where a polarity contrast takes place. However, important bottlenecks here are instances where a hashtag is required to understand the irony (and which will, consequently always be impossible to recognise without additionnal context), and polarity contrasts that involve implicit sentiment.

Indeed, experiments with a contrast-based system for irony revealed that, even when implicit sentiment is identified, finding the contrasting explicit sentiment using a lexicon-based approach is sometimes challenging (e.g.'[...] is exactly what I need'). Moreover, detecting explicit sentiment contrasts as a clue for irony is prone to overgeneration, as non-ironic tweets are also likely to contain contrastive evaluations (e.g. when discussing positive and negative aspects of something). Here as well, improving the sentiment lexicon-based approach or adopting supervised sentiment analysis might enhance the results, hence this will constitute an important direction for future work.

While our annotation guidelines distinguish between different types of irony on Twitter, in the current experiments we approach irony detection as a binary classification task. This way, this thesis provides insights into the feasibility of irony detection in general. Moreover, not distinguishing between different types of irony also allows to compare our approach with the state of the art. When applied for specific purposes (e.g. improving automatic sentiment analysis), however, a fine-grained classification might be worthwhile to detect cases of irony where a polarity inversion takes place. Nevertheless, we believe more data would be necessary to do such fine-grained classification, but also to enhance performance of the binary classifier, as related work often obtains similar results with a less complex model, but much more data.

Finally, the presented approach is language-independent, provided that annotated data are available. As such, we aim to perform similar experiments on a Dutch dataset that is currently under construction.

8.3 Modelling implicit sentiment

> → *Research question 2: Is it feasible to automatically detect implicit or prototypical sentiment related to particular situations and does it benefit automatic irony detection?*

Our second research question addresses the automatic definition of implicit sentiment, also referred to as *prototypical sentiment* (Hoste et al. 2016) and *connotative knowledge* or *sentics* (Cambria et al. 2016).

The annotation of our irony corpus revealed that many instances of irony are realised through a polarity contrast between explicit sentiment expressions and *targets* carrying implicit sentiment (e.g. 'Gawd! I love 9am lectures'). Our irony detection experiments in Chapter 4 revealed that false negatives of the system often include such implicit polarity contrasts, suggesting that implicit sentiment recognition has the potential to improve classification performance.

We investigated how implicit or prototypical sentiment can be inferred automatically. As the result of our manual irony annotations, we have at our disposal a set of 671 concepts (or *targets*) linked to their implicit sentiment. Using these manual annotations as gold standard, we automatically determined the implicit sentiment linked to these targets. For this purpose, we explored two methods: i) SenticNet 4, a state-of-the-art knowledge base (Cambria et al. 2016), and ii) a data-driven approach based on real time Twitter data.

Related work by Riloff et al. (2013) involves a bootstrapping approach to learn positive and negative situation (i.e. verb) phrases in the vicinity of positive seed words like 'love', 'enjoy' and 'hate'. The researchers showed that using implicit sentiment information benefits irony detection, however, the learned implicit sentiment phrases were restricted to verb phrases. In this research, we worked the other way around and started with manually annotated prototypical situations and concepts like '9am lectures', 'working in the weekend', 'up all night 2 nights in a row' and tried to infer their prototypical sentiment automatically. As such, while Riloff's (2013) main challenge was to extract more specific situations than verb phrases, ours was to find a good abstraction method to enable an efficient look-up of the phrases in SenticNet and by using Twitter.

SenticNet

Using a knowledge base to infer implicit sentiment, we looked up the SenticNet polarity of the target or calculated its overall polarity based on the scores of its individual words. We compared the suitability of the approach when looking up the original tokens in the targets, content word tokens and semantic concepts (Rajagopal et al. 2013) (e.g. 'feeling ill, banging head...' → 'feel_ill', 'bang_head'). The experiments revealed that the third method is the most effective, as it protects some *semantic atoms* (Cambria and Hussain 2015) which lose their original meaning when broken down into single words.

Overall, the experiments revealed that implicit sentiment seldom applies to words or phrases in isolation. In fact, many concepts are non-compositional, meaning that their meaning cannot merely be derived from the meaning of their constituent words in isolation. Although we were able to correctly determine the implicit sentiment related to 37% of the targets using a multiword-based lookup, the current method is perhaps too naive an approach to model implicit sentiment linked to a concept, as it often breaks the latter down into single words, which consequently lose their prior meaning.

Twitter

For our second approach to modelling implicit sentiment, we made use of Twitter, hypothesising that it provides valuable insights into people's opinions and hence allows to infer implicit sentiment related to particular concepts and situations.

To test this hypothesis, we investigated whether a large number of explicit opinions about a concept or situation is a reliable indication for the implicit or connotative sentiment related to that situation. Concretely, for each of the targets, a maximum of 500 related tweets were crawled using the Twitter API. Next, supervised sentiment analysis was applied to determine the overall sentiment in these tweets. As collecting sufficient tweets is crucial and some concepts are very specific (e.g. 'Christmas shopping on 2hrs sleep'), we explored a number of strategies to increase coverage, including looking for i) content words, ii) dependency heads and iii) verb-object patterns in the concepts, rather than the original tokens.

The experiments revealed that analysing tweets about a concept or situation is a viable method to determine the implicit sentiment related to that concept or situation. In effect, approximately 70% of all targets were assigned a correct implicit sentiment, which is a considerable improvement compared to the SenticNet-based approach. Experimenting with different abstraction methods revealed that, although more tweets were found when searching concepts based on content words or dependency heads, this had no substantial effect on the system's performance on defining implicit sentiment.

In sum, using Twitter to find the implicit sentiment related to particular concepts certainly has a number of advantages compared to the SenticNet-based approach. First, the former approach allows to derive sentiment for the entire concept, instead of breaking it down into individual words or multiwords. Second, by applying sentiment to an entire tweet, the approach takes context into account. Third, the method allows to collect real time data, making it possible to analyse opinions on topical concepts even before they could be inserted into knowledge bases.

Limitations and future work

Some challenges we observed when using SenticNet to determine implicit sentiment are the lack of coverage of some words in the database on the one hand, and confusing polarity values in the database (e.g. 'talk': -0.85) on the other hand. While including complex normalisation (Schulz et al. 2016) as a preprocessing step could increase coverage, we believe that the major drawbacks of the approach in general are its inability to preserve semantic atoms (unless they are included as multiword terms in the knowledge base), the lack of context, which is crucial in sentiment analysis, and its static character compared to Twitter.

While using Twitter to infer implicit sentiment clearly shows some advantages compared to the knowledge base approach, a number of important limitations were identified. First, modelling prototypical sentiment based on Twitter requires an automatic sentiment analysis system and imposes the collection of sufficient data. This brings us to the second drawback of the approach, namely the collection of tweets, which is hindered by the search constraints of the Twitter API and depends on the complexity of the search query.

While Riloff et al. (2013) concluded that most situation phrases carrying implicit sentiment are verb phrases, we observed that only 40% of the targets contained a verb-object pattern, suggesting that by focussing on verb phrases only, many other implicit sentiment concepts (e.g. '8.30 a.m. conference calls', 'no sleep 2 days in a row') are not taken into account, although these are common realisations of implicit sentiment as well. It will therefore be interesting to explore the coverage of our targets on Twitter when reducing them to verb and noun phrases, and to examine whether they are still specific enough (i.e. without losing much semantic information).

8.4 Sentiment analysis use case

> → *Research question 3: Can automatic irony detection enhance state-of-the-art sentiment analysis?*

As mentioned several times throughout this thesis, sentiment classifiers have proven to struggle with ironic text (Maynard and Greenwood 2014). Hence, to address the last research question, Chapter 7 presents an extrinsic evaluation of our irony classifier by testing its potential to improve state-of-the-art sentiment analysis.

The sentiment classifier we described in the chapter uses a support vector machine (SVM) and was trained on English tweets provided within the framework of the SemEval-2014 task *Sentiment analysis in Twitter* 9 (Rosenthal et al. 2014). The system exploits a rich and varied feature set and the model was optimised using feature selection and hyperparameter optimisation. No specific irony features were included in the model. To investigate the impact of irony, the optimised sentiment model was validated on two SemEval test sets containing

100% (*SemEval2014Sarcasm*) and 50% (*SemEval2015Sarcasm-balanced*) ironic tweets, before and after the classifier was informed by our irony detection system (SVM+clash).

The results of the experiments showed that sentiment classification clearly benefits from automatic irony detection, showing a performance increase of 20% to 40% on the SemEval2014Sarcasm (i.e. 100% ironic) corpus, and 7% to 12% on the balanced corpus. Analysis revealed that the sentiment classifier suffers from the class imbalance in the ironic datasets (i.e. containing mostly negative tweets), which caused it to overgenerate positive and neutral class instances. It was also observed that respectively 29% and 50% of the classification errors in the non-balanced and balanced corpus are due to errors percolation from the irony detection step.

Limitations and future work

A qualitative analysis revealed that the hybrid sentiment analyser performs best on *ironic by clash* instances, given that other instances of irony do not require the explicit polarity to be inverted. In effect, when identified as ironic, their polarity was also inverted, although this was mostly unnecessary. Fine-grained irony detection might be helpful to tackle this problem. This way, whether a sentiment prediction should be inverted or not would depend on the type of irony that is detected (i.e. *ironic by clash*).

As mentioned earlier in this chapter, other important research directions include optimisation of the irony detection system by means of feature filtering and hyperparameter tuning, as similar experiments on the sentiment classifier have proven worthwhile. Moreover, orthographic normalisation of noisy text as a preprocessing step could further enhance the sentiment classifier we currently have in place (see Chapter 7).

Bibliography

Al-Mannai, K., Alshikhabobakr, H., Bin Wasi, S., Neyaz, R., Bouamor, H. and Mohit, B.: 2014, CMUQ-Hybrid: Sentiment Classification By Feature Engineering and Parameter Tuning, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Association for Computational Linguistics and Dublin City University, pp. 181–185.

Attardo, S.: 2000, Irony as relevant inappropriateness, *Journal of Pragmatics* **32**(6), 793–826.

Balahur, A., Hermida, J. M., Montoyo, A. and Muñoz, R.: 2011, EmotiNet: A Knowledge Base for Emotion Detection in Text Built on the Appraisal Theories, *Proceedings of the 16th International Conference on Natural Language Processing and Information Systems*, NLDB'11, Springer-Verlag, Berlin, Heidelberg, pp. 27–39.

Balahur, A. and Tanev, H.: 2016, Detecting Implicit Expressions of Affect from Text using Semantic Knowledge on Common Concept Properties, *in* N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, B. Mariani, H. Mazo, A. Moreno, J. Odijk and S. Piperidis (eds), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, European Language Resources Association (ELRA), Paris, France.

Bamman, D. and Smith, N. A.: 2015, Contextualized Sarcasm Detection on Twitter, *Proceedings of the Ninth International Conference on Web and Social Media, (ICWSM)*, AAAI Press, Oxford, UK, pp. 574–577.

Barbieri, F., Ronzano, F. and Saggion, H.: 2014, Italian irony detection in twitter: a first approach, *The First Italian Conference on Computational Linguistics CLiC-it 2014*, pp. 28–32.

Barbieri, F. and Saggion, H.: 2014, Modelling Irony in Twitter, *Proceedings of the EACL Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL'14)*, Association for Computational Linguistics, pp. 56–64.

Barthi, S. K., Vacha, B., Pradhan, R., Babu, K. S. and Jena, S. K.: 2016, Sarcastic sentiment detection in tweets streamed in real time: a big data approach, *Digital Communications and Networks* **2**(3), 108–121.

Bin Wasi, S., Neyaz, R., Bouamor, H. and Mohit, B.: 2014, CMUQ@Qatar:Using Rich Lexical Features for Sentiment Analysis on Twitter, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Association for Computational Linguistics and Dublin City University, pp. 186–191.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K. and Zampieri, M.: 2016, Findings of the 2016 Conference on Machine Translation (WMT16), *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, Association for Computational Linguistics, pp. 131–198.

Bosco, C., Patti, V. and Bolioli, A.: 2013, Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT, *IEEE Intelligent Systems* **28**(2), 55–63.

Bouazizi, M. and Ohtsuki, T.: 2015, Opinion Mining in Twitter How to Make Use of Sarcasm to Enhance Sentiment Analysis, *in* J. Pei, F. Silvestri and J. Tang (eds), *ASONAM*, Association for Computing Machinery, pp. 1594–1597.

Bouazizi, M. and Ohtsuki, T.: 2016, Sarcasm detection in twitter: "all your products are incredibly amazing!!!" - are they really?, *2015 IEEE Global Communications Conference, GLOBECOM 2015*, Institute of Electrical and Electronics Engineers Inc., pp. 1–6.

Brown, P. and Levinson, S. C.: 1987, *Politeness: Some Universals in Language Usage*, Studies in Interactional Sociolinguistics, Cambridge University Press.

Bryant, G. A. and Fox Tree, J. E.: 2002, Recognizing verbal irony in spontaneous speech, *Metaphor and symbol* **17**(2), 99–119.

Burgers, C.: 2010, *Verbal irony: Use and effects in written discourse*, PhD thesis, UB Nijmegen [Host].

Buschmeier, K., Cimiano, P. and Klinger, R.: 2014, An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews, *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Association for Computational Linguistics, Baltimore, Maryland, pp. 42–49.

Cambria, E., Das, D., Bandyopadhyay, S. and Feraco, A. (eds): 2017, *A Practical Guide to Sentiment Analysis*, Springer International Publishing, Cham, Switzerland.

Cambria, E. and Hussain, A.: 2015, *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*, 1st edn, Springer Publishing Company, Incorporated.

Cambria, E., Hussain, A., Havasi, C. and Eckl, C.: 2009, Common Sense Computing: From the Society of Mind to Digital Intuition and beyond, *in* J. Fierrez, J. Ortega-Garcia, A. Esposito, A. Drygajlo and M. Faundez-Zanuy (eds), *Biometric ID Management and Multimodal Communication: Joint COST 2101 and 2102 International Conference, BioID_MultiComm 2009*, Springer Berlin Heidelberg, Madrid, Spain, pp. 252–259.

Cambria, E., Poria, S., Bajpai, R. and Schuller, B.: 2016, SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives, *in* N. Calzolari, Y. Matsumoto and R. Prasad (eds), *Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, Association for Computational Linguistics, Osaka, Japan, pp. 2666–2677.

Cambria, E., Speer, R., Havasi, C. and Hussain, A.: 2010, SenticNet: A Publicly Available Semantic Resource for Opinion Mining, *AAAI Fall Symposium: Commonsense Knowledge (AAAI Technical Report)*, Vol. FS-10-02, AAAI.

Camp, E.: 2012, Sarcasm, Pretense, and The Semantics/Pragmatics Distinction, *Nous* **46**(4), 587–634.

Carletta, J.: 1996, Assessing Agreement on Classification Tasks: The Kappa Statistic, *Computational Linguistics* **22**(2), 249–254.

Carvalho, P., Sarmento, L., Silva, M. J. and de Oliveira, E.: 2009, Clues for Detecting Irony in User-generated Contents: Oh...!! It's "So Easy" ;-), *Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA '09, ACM, New York, NY, USA, pp. 53–56.

Chang, C.-C. and Lin, C.-J.: 2011, LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology* **2**(3), 27:1–27:27.

Choi, J. D., Tetreault, J. R. and Stent, A.: 2015, It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool, *ACL (1)*, Association for Computer Linguistics, pp. 387–396.

Clark, H. H. and Gerrig, R. J.: 1984, On the pretense theory of irony, *Journal of Experimental Psychology: General* **113**(1), 121–126.

Clift, R.: 1999, Irony in Conversation, *Language in Society* **28**(4), 523–553.

Cortes, C. and Vapnik, V.: 1995, Support-Vector Networks, *Machine Learning* **20**(3), 273–297.

Coulson, S.: 2005, Sarcasm and the Space Structuring Model, *in* S. Coulson and B. Lewandowska-Tomaszczyk (eds), *The Literal and Nonliteral in Language and Thought*, Peter Lang, pp. 129–144.

Currie, G.: 2006, Why irony is pretence, *in* S. Nichols (ed.), *The Architecture of the Imagination: New Essays on Pretence, Possibility, and Fiction*, Clarendon Press.

Dadvar, M.: 2014, *Experts and machines united against cyberbullying*, Phd thesis, University of Twente.

Daelemans, W., Zavrel, J., van der Sloot, K. and van den Bosch, A.: 2009, TiMBL: Tilburg Memory Based Learner, version 6.2, Reference Guide, *Technical Report 09-01*, ILK Research Group.

Davidov, D., Tsur, O. and Rappoport, A.: 2010, Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon, *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL'10)*, Association for Computational Linguistics, Uppsala, Sweden, pp. 107–116.

de Marneffe, M.-C., MacCartney, B. and Manning, C. D.: 2006, Generating Typed Dependency Parses from Phrase Structure Parses, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)*, pp. 449–454.

Deriu, J., Gonzenbach, M., Uzdilli, F., Lucchi, A., De Luca, V. and Jaggi, M.: 2016, SwissCheese at SemEval-2016 Task 4: Sentiment Classification Using an Ensemble of Convolutional Neural Networks with Distant Supervision, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Association for Computational Linguistics, San Diego, California, pp. 1124–1128.

Desmet, B., Hoste, V., Verstraeten, D. and Verhasselt, J.: 2013, Gallop Documentation, *Technical Report LT3 13-03*, University College Ghent.

Dinakar, K., Jones, B., Havasi, C., Lieberman, H. and Picard, R.: 2012, Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying, *ACM Transactions on Interactive Intelligent Systems* **2**(3), 18:1–18:30.

Eisterhold, J., Attardo, S. and Boxer, D.: 2006, Reactions to irony in discourse: evidence for the least disruption principle, *Journal of Pragmatics* **38**(8), 1239–1256.

Esuli, A. and Sebastiani, F.: 2006, SENTIWORDNET: A publicly available lexical resource for opinion mining, *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06*, pp. 417–422.

Farías, D. I. H., Patti, V. and Rosso, P.: 2016, Irony detection in twitter: The role of affective content, *ACM Transactions on Internet Technology* **16**(3), 19:1–19:24.

Fernandes, J. A., Irigoien, X., Goikoetxea, N., Lozano, J. A., Inza, I. n., Pérez, A. and Bode, A.: 2010, Fish recruitment prediction using robust supervised classification methods, *Ecological Modelling* **221**.

Filatova, E.: 2012, Irony and sarcasm: Corpus generation and analysis using crowdsourcing, *in* N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk and S. Piperidis (eds), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association, Istanbul, Turkey, pp. 392–398.

Fillmore, C. J., Johnson, C. R. and Petruck, M. R.: 2003, Background to Framenet, *International Journal of Lexicography* **16**(3), 235–250.

Fleiss, J. L.: 1971, Measuring nominal scale agreement among many raters, *Psychological Bulletin* **76**(5), 378–382.

Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J. and Reyes, A.: 2015, SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval'15)*, Association for Computational Linguistics, Denver, Colorado, pp. 470–478.

Ghosh, A. and Veale, T.: 2016, Fracking Sarcasm using Neural Network, *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Association for Computational Linguistics, San Diego, California, pp. 161–169.

Gibbs, R. W.: 1986, On the psycholinguistics of sarcasm, *Journal of Experimental Psychology* **115**(1), 3–15.

Gibbs, R. W., O'Brien, J. E. and Doolittle, S.: 1995, Inferring meanings that are not intended: Speakers' intentions and irony comprehension, *Discourse Processes* **20**(2), 187–203.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J. and Smith, N. A.: 2011, Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT'11)*, Association for Computational Linguistics, Portland, Oregon, pp. 42–47.

Giora, R.: 1995, On irony and negation, *Discourse Processes* **19**(2), 239–264.

Giora, R., Fein, O., Ganzi, J., Levi, N. A. and Sabah, H.: 2005, On negation as mitigation: The case of negative irony, *Discourse Processes* **39**(1), 81–100.

González-Ibáñez, R., Muresan, S. and Wacholder, N.: 2011, Identifying Sarcasm in Twitter: A Closer Look, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT'11)*, Association for Computational Linguistics, Portland, Oregon, pp. 581–586.

Grice, H. P.: 1975, Logic and Conversation, *in* P. Cole and J. L. Morgan (eds), *Syntax and Semantics: Vol. 3*, Academic Press, New York, pp. 41–58.

Grice, H. P.: 1978, Further Notes on Logic and Conversation, *in* P. Cole (ed.), *Syntax and Semantics: Vol. 9*, Academic Press, New York, pp. 113–127.

Günther, T., Vancoppenolle, J. and Johansson, R.: 2014, RTRGO: Enhancing the GU-MLT-LT System for Sentiment Analysis of Short Messages, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pp. 497–502.

Gupta, R. K. and Yang, Y.: 2017, CrystalNest at SemEval-2017 Task 4: Using Sarcasm Detection for Enhancing Sentiment Classification and Quantification, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Association for Computational Linguistics, Vancouver, Canada, pp. 617–624.

Haiman, J.: 1998, *Talk Is Cheap : Sarcasm, Alienation, and the Evolution of Language: Sarcasm, Alienation, and the Evolution of Language*, Oxford University Press, USA.

Hallmann, K., Kunneman, F., Liebrecht, C., van den Bosch, A. and van Mulken, M.: 2016a, Sarcastic Soulmates: Intimacy and irony markers in social media messaging, *LiLT (Linguistic Issues in Language Technology)* **14**.

Hallmann, K., Kunneman, F., Liebrecht, C., van den Bosch, A. and van Mulken, M.: 2016b, Signaling sarcasm: From hyperbole to hashtag, *Linguistic Issues in Language Technology* **14**(7), 500–509.

Heafield, K., Pouzyrevsky, I., Clark, J. H. and Koehn, P.: 2013, Scalable Modified Kneser-Ney Language Model Estimation, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, pp. 690–696.

Hogenboom, A., Bal, D., Frasincar, F., Bal, M., De Jong, F. and Kaymak, U.: 2015, Exploiting Emoticons in Polarity Classification of Text, *Journal of Web Engineering* **14**(1-2), 22–40.

Hoste, V., Lefever, E., van der Waart van Gulik, S. and Desmet, B.: 2016, TripleSent: a triple store of events associated with their prototypical sentiment, *Proceedings of The Eighth International Conference on Information, Process, and Knowledge Management*, IARIA, pp. 91–93.

Hsu, C.-W., Chang, C.-C. and Lin, C.-J.: 2003, A Practical Guide to Support Vector Classification, *Technical report*, Department of Computer Science, National Taiwan University.
URL: *http://www.csie.ntu.edu.tw/ cjlin/papers.html*

Jasso López, G. and Meza Ruiz, I.: 2016, Character and Word Baselines Systems for Irony Detection in Spanish Short Texts, *Procesamiento de Lenguaje Natural* **56**, 41–48.

Joshi, A., Bhattacharyya, P. and Carman, M. J.: 2016, Automatic Sarcasm Detection:A Survey, *CoRR* **abs/1602.03426**.

Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P. and Carman, M. J.: 2016, Are Word Embedding-based Features Useful for Sarcasm Detection?, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, Association for Computational Linguistics, Austin, Texas, pp. 1006–1011.

Joshi, M. and Penstein-Rosé, C.: 2009, Generalizing Dependency Features for Opinion Mining, *Proceedings of the ACL-IJCNLP 2009 Conference*, Association for Computational Linguistics, pp. 313–316.

Karoui, J., Benamara, F., Moriceau, V., Aussenac-Gilles, N. and Hadrich Belguith, L.: 2015, Towards a Contextual Pragmatic Model to Detect Irony in

Tweets, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, Association for Computational Linguistics, Beijing, China, pp. 644–650.

Karoui, J., Benamara, F., Moriceau, V., Patti, V., Bosco, C. and Aussenac-Gilles, N.: 2017, Exploring the Impact of Pragmatic Phenomena on Irony Detection in Tweets: A Multilingual Corpus Study, *EACL-European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Valencia, Spain, pp. 262–272.

Keerthi, S. S. and Lin, C.-J.: 2003, Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel, *Neural Computation* **15**(7), 1667–1689.

Kennedy, A. and Inkpen, D.: 2006, Sentiment Classification of Movie Reviews Using Contextual Valence Shifters, *Computational Intelligence* **22**(2), 110–125.

Kihara, Y.: 2005, The Mental Space Structure of Verbal Irony, *Cognitive Linguistics* **16**(3).

Kökciyan, N., Çelebi, A., Özgür, A. and Üsküdarli, S.: 2013, BOUNCE: Sentiment Classification in Twitter using Rich Feature Sets, *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, Association for Computational Linguistics, Atlanta, Georgia, pp. 554–561.

Kralj Novak, P., Smailović, J., Sluban, B. and Mozetič, I.: 2015, Sentiment of Emojis, *PLOS ONE* **10**(12), 1–22.

Kreuz, R. J. and Glucksberg, S.: 1989, How to be sarcastic: The echoic reminder theory of verbal irony, *Journal of Experimental Psychology: General* **118**(4), 374.

Kreuz, R. J. and Roberts, R. M.: 1993, On satire and parody: The importance of being ironic, *Metaphor and Symbol* **8**(2), 97–109.

Kumon-Nakamura, S., Glucksberg, S. and Brown, M.: 1995, How About Another Piece of Pie: The Allusional Pretense Theory of Discourse Irony, *Journal of Experimental Psychology: General* **124**(1), 3.

Kunneman, F., Liebrecht, C., van Mulken, M. and van den Bosch, A.: 2015, Signaling sarcasm: From hyperbole to hashtag, *Information Processing & Management* **51**(4), 500–509.

Landis, J. R. and Koch, G. G.: 1977, The measurement of observer agreement for categorical data, *Biometrics* **33**(1).

Lee, C. J. and Katz, A. N.: 1998a, The differential role of ridicule in sarcasm and irony, *Metaphor and symbol* **13**(1), 1–15.

Lee, C. J. and Katz, A. N.: 1998b, The Differential Role of Ridicule in Sarcasm and Irony, *Metaphor and Symbol* **13**(1), 1–15.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S. and Bizer, C.: 2015, DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic Web* **6**(2), 167–195.

Lenat, D. B.: 1995, CYC: A Large-scale Investment in Knowledge Infrastructure, *Communications of the ACM* **38**(11), 33–38.

Liebrecht, C., Kunneman, F. and van den Bosch, A.: 2013, The perfect solution for detecting sarcasm in tweets #not, *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA'13)*, Association for Computational Linguistics, Atlanta, Georgia, pp. 29–37.

Liu, B.: 2012, *Sentiment Analysis and Opinion Mining*, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers.

Liu, B., Hu, M. and Cheng, J.: 2005, Opinion Observer: Analyzing and Comparing Opinions on the Web, *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*, Association for Computing Machinery, Chiba, Japan, pp. 342–351.

Lucariello, J.: 1994, Situational Irony: A Concept of Events Gone Awry., *Journal of Experimental Psychology: General* **123**(2), 129–145.

Lunando, E. and Purwarianti, A.: 2013, Indonesian social media sentiment analysis with sarcasm detection, *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, IEEE, pp. 195–198.

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

Maynard, D. and Greenwood, M.: 2014, Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis, *in* N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk and S. Piperidis (eds), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, European Language Resources Association, Reykjavik, Iceland.

McQuarrie, E. F. and Mick, D. G.: 1996, Figures of Rhetoric in Advertising Language, *The Journal of Consumer Research* **22**(4), 424–438.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: 2013, Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems*, pp. 3111–3119.

Miller, G. A.: 1995, WordNet: A Lexical Database for English, *Communications of the ACM* **38**(11), 39–41.

Miura, Y., Sakaki, S., Hattori, K. and Ohkuma, T.: 2014, TeamX: A Sentiment Analyzer with Enhanced Lexicon Mapping and Weighting Scheme for Unbalanced Data, *Proceedings of the International Workshop on Semantic Evaluation (SemEval 2014)*, Association for Computational Linguistics and Dublin City University, pp. 628–632.

Mohammad, S. M., Sobhani, P. and Kiritchenko, S.: 2016, Stance and Sentiment in Tweets, *CoRR* **abs/1605.01655**.

Mohammad, S. M. and Turney, P. D.: 2013, Crowdsourcing a Word-Emotion Association Lexicon, **29**(3), 436–465.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F. and Stoyanov, V.: 2016, SemEval-2016 Task 4: Sentiment Analysis in Twitter, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Association for Computational Linguistics, San Diego, California, pp. 1–18.

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A. and Wilson, T.: 2013, SemEval 2013 Task 2: Sentiment Analysis in Twitter, *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, ACL, pp. 312–320.

Nielsen, F. Å.: 2011, A new ANEW: evaluation of a word list for sentiment analysis in microblogs, *in* M. Rowe, M. Stankovic, A.-S. Dadzie and M. Hardey (eds), *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, Vol. 718, pp. 93–98.

Noreen, E. W.: 1989, *Computer Intensive Methods for Testing Hypothesis: An Introduction*, John Wiley & Sons, New York.

Nunberg, G.: 2001, *The Way we Talk Now: Commentaries on Language and Culture*, Houghton Mifflin.

Page, R.: 2012, The linguistics of self-branding and micro-celebrity in Twitter: The role of hashtags, *Discourse & Communication* **6**(2), 181–201.

Pang, B. and Lee, L.: 2008, Opinion Mining and Sentiment Analysis, *Foundations and Trends in Information Retrieval* **2**(1-2), 1–135.

Peng, C.-C., Lakis, M. and Wei Pan, J.: 2015, Detecting Sarcasm in Text: An Obvious Solution to a Trivial Problem, *Stanford CS 229 Machine Learning Final Projects 2015*.

Pennebaker, J. W., Francis, M. E. and Booth, R. J.: 2001, *Linguistic Inquiry and Word Count: LIWC 2001*, Lawrence Erlbaum Associates, Mahwah, NJ.

Polanyi, L. and Zaenen, A.: 2006, Contextual Valence Shifters, *in* J. G. Shanahan, Y. Qu and J. Wiebe (eds), *Computing Attitude and Affect in Text: Theory and Applications*, Springer Netherlands, Dordrecht, pp. 1–10.

Poria, S., Cambria, E., Hazarika, D. and Vij, P.: 2016, A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, Osaka, Japan, pp. 1601–1612.

Poria, S., Cambria, E., Winterstein, G. and Huang, G.-B.: 2014, Sentic patterns: Dependency-based rules for concept-level sentiment analysis, *Knowledge-Based Systems* **69**, 45–63.

Ptáček, T., Habernal, I. and Hong, J.: 2014, Sarcasm detection on czech and english twitter, *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pp. 213–223.

Quintiliano, M. F. and Butler, H. E.: 1959, *The Institutio oratoria of Quintilian*, London: Wiliam Heinemann.

Rajagopal, D., Cambria, E., Olsher, D. and Kwok, K.: 2013, A Graph-based Approach to Commonsense Concept Extraction and Semantic Similarity Detection, *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*, Association for Computing Machinery, Rio de Janeiro, Brazil, pp. 565–570.

Reyes, A. and Rosso, P.: 2012, Making objective decisions from subjective data: Detecting irony in customer reviews, *Decision Support Systems* **53**(4), 754–760.

Reyes, A., Rosso, P. and Veale, T.: 2013, A Multidimensional Approach for Detecting Irony in Twitter, *Language Resources and Evaluation* **47**(1), 239–268.

Riloff, E., Qadir, A., Surve, P., Silva, L. D., Gilbert, N. and Huang, R.: 2013, Sarcasm as Contrast between a Positive Sentiment and Negative Situation, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, Association for Computational Linguistics, Seattle, Washington, USA, pp. 704–714.

Ritchie, D.: 2005, Frame-Shifting in Humor and Irony, *Metaphor and Symbol* **20**(4), 275–294.

Ritter, A., Clark, S., Mausam and Etzioni, O.: 2011, Named entity recognition in tweets: An experimental study., *Proceedings of Empirical Methods for Natural Language Processing EMNLP*, pp. 1524–1534.

Rockwell, P.: 2000, Lower, Slower, Louder: Vocal Cues of Sarcasm, *Journal of Psycholinguistic Research* **29**(5), 483–495.

Rosenthal, S., Farra, N. and Nakov, P.: 2017, SemEval 2017 Task 4: Sentiment Analysis in Twitter, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, ACL, Vancouver, Canada, pp. 502–518.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A. and Stoyanov, V.: 2015, SemEval 2015 Task 10: Sentiment Analysis in Twitter, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, ACL, pp. 451–463.

Rosenthal, S., Ritter, A., Nakov, P. and Stoyanov, V.: 2014, SemEval-2014 Task 9: Sentiment Analysis in Twitter, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pp. 73–80.

Schulz, S., Pauw, G. D., Clercq, O. D., Desmet, B., Hoste, V., Daelemans, W. and Macken, L.: 2016, Multimodular Text Normalization of Dutch User-Generated Content, *ACM Transactions on Intelligent Systems and Technology* **7**(4), 61:1–61:22.

Shelley, C.: 2001, The bicoherence theory of situational irony, *Cognitive Science* **25**(5), 775–818.

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T. and Zhu, W. L.: 2002, Open Mind Common Sense: Knowledge Acquisition from the General Public, *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, Springer-Verlag, London, UK, UK, pp. 1223–1237.

Sokolova, M. and Lapalme, G.: 2009, A systematic analysis of performance measures for classification tasks, *Information Processing & Management* **45**(4), 427–437.

Speer, R. and Havasi, C.: 2013, ConceptNet 5: A Large Semantic Network for Relational Knowledge, *in* I. Gurevych and J. Kim (eds), *The People's Web Meets NLP: Collaboratively Constructed Language Resources*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 161–176.

Sperber, D. and Wilson, D.: 1981, Irony and the Use - Mention Distinction, *in* P. Cole (ed.), *Radical Pragmatics*, Academic Press, New York, pp. 295–318.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S. and Tsujii, J.: 2012, BRAT: A Web-based Tool for NLP-assisted Text Annotation, *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL'12)*, Association for Computational Linguistics, Avignon, France, pp. 102–107.

Stone, P. J., Dunphy, D. C., Smith, M. S. and Ogilvie, D. M.: 1966, The General Inquirer: A Computer Approach to Content Analysis, *MIT Press* .

Stranisci, M., Bosco, C., Farías, D. I. H. and Patti, V.: 2016, Annotating Sentiment and Irony in the Online Italian Political Debate on #labuonascuola, *in* N. C. C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk and S. Piperidis (eds), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, European Language Resources Association (ELRA), Paris, France.

Tang, D., Wei, F., Qin, B., Liu, T. and Zhou, M.: 2014, Coooolll: A Deep Learning System for Twitter Sentiment Classification, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pp. 208–212.

Tsur, O., Davidov, D. and Rappoport, A.: 2010, ICWSM-A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews, *ICWSM*.

Turing, A. M.: 1950, Computing Machinery and Intelligence, *Mind* **49**, 433–460.

Utsumi, A.: 2000, Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony, *Journal of Pragmatics* **32**(12), 1777–1806.

Van de Kauter, M., Coorman, G., Lefever, E., Desmet, B., Macken, L. and Hoste, V.: 2013, LeTs Preprocess: The multilingual LT3 linguistic preprocessing toolkit, *Computational Linguistics in the Netherlands Journal* **3**, 103–120.

Van de Kauter, M., Desmet, B. and Hoste, V.: 2015, The good, the bad and the implicit: a comprehensive approach to annotating explicit and implicit sentiment, *Language Resources and Evaluation* **49**(3), 685–720.

Van Hee, C., Lefever, E. and Hoste, V.: 2015, LT3: Sentiment Analysis of Figurative Tweets: piece of cake #NotReally, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Association for Computational Linguistics, Denver, Colorado, pp. 684–688.

Van Hee, C., Lefever, E. and Hoste, V.: 2016a, Guidelines for Annotating Irony in Social Media Text, version 2.0, *Technical Report 16-01*, LT3, Language and Translation Technology Team–Ghent University.

Van Hee, C., Lefever, E. and Hoste, V.: 2016b, Monday mornings are my fave : #not Exploring the Automatic Recognition of Irony in English tweets, *Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, Osaka, Japan, pp. 2730–2739.

Van Hee, C., Lefever, E. and Hoste, V.: 2016c, Exploring the realization of irony in Twitter data, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, European Language Resources Association (ELRA), Portorož, Slovenia, pp. 1795–1799.

Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W. and Hoste, V.: 2015, Detection and fine-grained classification of cyberbullying events, *in* G. Angelova, K. Bontcheva and R. Mitkov (eds), *Proceedings of Recent Advances in Natural Language Processing, Proceedings*, Hissar, Bulgaria, pp. 672–680.

Van Hee, C., Van de Kauter, M., De Clercq, O., Lefever, E., Desmet, B. and Hoste, V.: Submitted, Noise or Music? Investigating the Usefulness of Normalisation for Robust Sentiment Analysis on Social Media Data.

Van Hee, C., Van de Kauter, M., De Clercq, O., Lefever, E. and Hoste, V.: 2014, LT3: Sentiment Classification in User-Generated Content Using a Rich Feature Set, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval'14)*, Association for Computational Linguistics, Dublin, Ireland, pp. 406–410.

Veale, T. and Hao, Y.: 2009, Support Structures for Linguistic Creativity: A Computational Analysis of Creative Irony in Similes, *in* N. Taatgen and H. van Rijn (eds), *Proceedings of CogSci 2009, the 31st Annual Meeting of the Cognitive Science Society*, Amsterdam, The Netherlands, pp. 1376–1381.

Vlastos, G.: 1987, Socratic irony, *The Classical Quarterly* **37**(1), 79–96.

Wallace, B. C.: 2015, Computational irony: A survey and new perspectives, *Artificial Intelligence Review* **43**(4), 467–483.

Wilson, D. and Sperber, D.: 1992, On verbal irony, *Lingua* **87**(1), 53–76.

Wilson, T., Wiebe, J. and Hoffmann, P.: 2005, Recognizing Contextual Polarity in Phrase-level Sentiment Analysis, *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT 2005)*, Association for Computational Linguistics, pp. 347–354.

Zhu, X., Kiritchenko, S. and Mohammad, S. M.: 2014, NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Association for Computational Linguistics and Dublin City University, pp. 443–447.

APPENDIX A

Publications

This appendix contains a list of all peer-reviewed journal and conference proceedings publications from the period 2014-2017.

- 2017
  - Van Hee, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, B., De Pauw, G., Daelemans, W. & Hoste, V. *Automatic Detection of Cyberbullying in Social Media Text.* PLOS ONE. Submitted on 06/02/2017.
  - Van Hee, C., Van de Kauter, M., De Clercq, O., Lefever, E., Desmet, B. & Hoste, V. *Noise or Music? Investigating the Usefulness of Normalisation for Robust Sentiment Analysis on Social Media Data.* Traitement Automatique des Langues. Accepted for publication with revisions.
  - Van Hee, C., Lefever, E. & Hoste, V. *Exploring the Fine-grained Analysis and Automatic Detection of Irony on Twitter.* Language Resources and Evaluation. Accepted for publication with revisions.

- 2016
  - Van Hee, C., Lefever, E. & Hoste, V. *Monday mornings are my fave: #not exploring the automatic recognition of Irony in English tweets.* In N. Calzolari, Y. Matsumoto, & R. Prasad (Eds.), Proceedings of COLING 2016, 26th International Conference on Computational Linguistics, 2730–2739, ACL.
  - Van Hee, C., Verleye, C. & Lefever, E. *Analysing emotions in social media coverage on Paris terror attacks: a pilot study.* In E. Lefever & D. J. Folds (Eds.), Proceedings of HUSO 2016, The Second International Conference on Human and Social Analytics, 33–37, IARIA.
  - Hoste, V., Van Hee, C. & Poels, K. *Towards a framework for the automatic detection of crisis emotions on social media : a corpus analysis of the tweets posted after the crash of Germanwings flight 9525.* Proceedings of HUSO 2016, The Second International Conference on Human and Social Analytics, 29–32, IARIA.
  - Van Hee, C., Lefever, E. & Hoste, V. *Exploring the realization of irony in Twitter data.* Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), 1795–1799, European Language Resources Association (ELRA).
  - Van Hee, C., Lefever, E. & Hoste, V. *Guidelines for Annotating Irony in Social Media Text*, version 2.0. LT3 Technical Report Series.
- 2015
  - Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W. & Hoste, V. Automatic detection and prevention of cyberbullying. In P. Lorenz & C. Bourret (Eds.), International Conference on Human and Social Analytics (HUSO 2015), 13–18, IARIA.
  - Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W. & Hoste, V. *Detection and fine-grained classification of cyberbullying events.* In G. Angelova, K. Bontcheva, & R. Mitkov (Eds.), Proceedings of Recent Advances in Natural Language Processing (RANLP 2015), 672–680.
  - Van Hee, C., Lefever, E. & Hoste, V. LT3: sentiment analysis of figurative tweets: piece of cake #NotReally. Proceedings of SemEval-2015, the 9th International Workshop on Semantic Evaluations, 684–688, Association for Computational Linguistics.
  - Van Hee, C., Verhoeven, B., Lefever, E., De Pauw, G., Daelemans, W. & Hoste, V. *Guidelines for the fine-grained analysis of cyberbullying*, version 1.0. LT3 Technical Report Series.

- 2014
  - Van Hee, C., Van de Kauter, M., De Clercq, O., Lefever, E. & Hoste, V. *LT3: sentiment classification in user-generated content using a rich feature set.* Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014), 406–410, Association for Computational Linguistics.
  - Desmet, B., De Clercq, O., Van de Kauter, M., Schulz, S., Van Hee, C. & Hoste, V. *Taaltechnologie 2.0: sentimentanalyse en normalisatie.* In Stefaan Evenepoel, P. Goethals, & L. Jooken (Eds.), Beschouwingen uit een talenhuis : opstellen over onderwijs en onderzoek in de vakgroep Vertalen, Tolken en Communicatie aangeboden aan Rita Godyns, 157–161, Ghent, Belgium: Academia Press.

APPENDIX B

Guidelines for annotating irony in social media text

Introduction

With the emergence of web 2.0, a large part of our daily communication has moved online. As a consequence, social media (e.g. Twitter, Facebook) have become a valuable source of information about the public's opinion for politicians, companies, researchers, trend watchers, and so on (Pang and Lee 2008). The past decade has seen an increased research interest in text mining on social media data. The frequent use of irony in this genre has important implications for tasks such as sentiment analysis and opinion mining (Maynard and Greenwood 2014, Reyes et al. 2013), which aim to extract positive and negative opinions automatically from online text. To develop or enhance sentiment analysis systems, or more broadly any task involving text interpretation (e.g. cyberbullying detection), it is of key importance to understand the linguistic realisation of irony, and to explore its automatic detection. Most computational approaches to date model irony by relying solely on categorical labels like irony hashtags (e.g. '#irony', '#sarcasm') assigned by the author of the text. To our knowledge, no guidelines presently exist for the more **fine-grained annotation** of irony in social media content without exploiting this hashtag information.

When describing how irony works, theorists traditionally distinguish between **situational irony** and **verbal irony**. Situational irony is often referred to as situations that fail to meet some expectations (Lucariello 1994, Shelley 2001). Shelley (2001) illustrates this with firefighters who have a fire in their kitchen while they are out to answer a fire alarm. Verbal irony is traditionally defined as expressions that convey an opposite meaning (e.g. Grice 1975, McQuarrie and Mick 1996, Quintiliano and Butler 1959) and implies the expression of a feeling, attitude or evaluation (Grice 1978, Wilson and Sperber 1992). There has been a large body of research in the past involving the definition of irony and the distinction between irony and sarcasm (Barbieri and Saggion 2014, Grice 1975, Kreuz and Glucksberg 1989, Wilson and Sperber 1992). To date, however, experts do not formally agree on the distinction between irony and sarcasm. For this reason, we elaborate a working definition that can cover both expressions described as verbal irony, and expressions described as sarcasm. In the definition, as well as in the remainder of this paper, we refer to this linguistic form as *irony.* In accordance with the traditional definition and that of Burgers (2010), we define irony as an *evaluative expression whose polarity (i.e., positive, negative) is inverted between the literal and the intended evaluation, resulting in an incongruence between the literal evaluation and its context.* More concretely, when speaking ironically, one expresses a positive sentiment whereas the implied sentiment is negative, or inversely.

In our working definition, no distinction is made between irony and sarcasm. However, the present annotation scheme allows to signal variants of verbal irony that are particularly **harsh** (i.e., carrying a mocking or ridiculing tone with the intention to hurt someone), since it has been considered a useful feature for distinguishing between irony and sarcasm (Barbieri and Saggion 2014, Lee and Katz 1998b).

In what follows, we present the different steps in the annotation of verbal irony in online text. All annotation steps can be executed using the brat rapid annotation tool (Stenetorp et al. 2012). The example sentences in the following sections are taken from a corpus of English tweets collected using the hashtags '#sarcasm', '#irony' and '#not'. It should be noted that not every element of the examples is annotated and discussed. We refer to Section B for detailed annotation examples.

Evaluative expressions

The present definition of irony is based on a polarity inversion between two evaluations. Annotators therefore look for expressions of an **evaluation** in the text under investigation. By an evaluation, we understand the entire text span by which someone or something (e.g. a product, an event, an organisation) is evaluated, including **modifiers** (see further). There are no restrictions as to which forms evaluations take; they can be verb phrases, predicative (adjective or nominal) expressions, emoticons, and so on. Nevertheless, when possible, annotators should include the verb and its apposition in the annotated text span of the evaluation, as well as modifiers (if present). Evaluative expressions can be found in examples 1 to 4.

> *(1) Oh how I love working in Baltimore #not*
> → 'Oh how I love' = **evaluation**

> *(2) What a shock. Duke Johnson is hurt in an important game. #sarcasm #canes*
> → 'What a shock' = **evaluation**

> *(3) So glad you'd rather read a book than acknowledge your own kid #not*
> → 'So glad' = **evaluation**

> *(4) Interesting visit with Terra Nova yesterday at Stoneleigh, class tent.*
> → 'Interesting' = **evaluation**
> → 'class' = **evaluation**

As shown in example 4, a text can include more than one evaluation.

Brat howto
In brat, if an evaluative expression consists of several non-consecutive parts (e.g. 'I love this band so much!'), the parts should be linked by means of drag and drop.

Evaluation polarity

An important subtask of annotating evaluations is **polarity assignment**, which involves determining whether the expressed evaluation is **positive** (e.g. 'love it!') or **negative** (e.g. 'it's a real nightmare'). It is possible that, due to ambiguity or a restricted context, it is not entirely clear whether an evaluation is positive or negative. Such evaluations receive the polarity label 'unknown'. Nevertheless, annotators should indicate a concrete polarity (i.e., positive or negative) as much as possible.

> *(5) I hate it when my mind keeps drifting to someone who no longer matters in life. #irony #dislike*
> → 'hate' = **evaluation** [negative polarity]
> → 'no longer matters in life' = **evaluation** [negative polarity]
> → '#dislike' = **evaluation** [negative polarity]

> *(6) First day off for summer...kids wake up at 6:01. Love them but not Awesome. #sleepisfortheweak #not.*
> → 'Love' = **evaluation** [positive polarity]
> → 'not Awesome' = **evaluation** [negative polarity]
> → '#sleepisfortheweak' = **evaluation** [negative polarity]

> *(7) I'm surprised you haven't been recruited by some undercover agency. #sarcasm*
> → ''m surprised' = **evaluation** [unknown polarity]

Like example 4, sentences 5 and 6 contain more than one evaluative expression. In Twitter data, hashtags may also contain evaluations. In this case, annotators are supposed to annotate the hashtag as an entire unit (including the hash sign '#'), even if it is a multiword expression (e.g. '#sleepisfortheweak').

Like words, all hashtags can be annotated, except '#sarcasm', '#irony' and '#not'. They were used to collect the data and are supposed to be left unannotated.

Modifiers

Sometimes, evaluative expressions are modified. This means that their polarity is changed by an element (i.e., a *modifier*) in the text. Modifiers are lexical items that cause a "**shift in the prior polarity** of other nearby lexical items" (Van de Kauter et al. 2015). They can be left out without losing the sentiment expression.

Two types of modifiers are distinguished in our annotation scheme: **(i) intensifiers**, which increase the intensity of the expressed sentiment and **(ii) diminishers**, which decrease the intensity of the expressed sentiment (Kennedy and

Inkpen 2006, Polanyi and Zaenen 2006). The evaluation polarity can be modified by adverbs (e.g. 'absolutely'), interjections (e.g. 'wow'), punctuation marks (e.g. '??!!'), emoticons, and so on. The modifiers in the sentences 8 and 9 are bold-faced.

> *(8) The* ***most*** *annoying kid lives next to my door****!!!***
> → 'most annoying' = **evaluation** [negative polarity]
> → 'most' = **intensifier** of *annoying*
> → '!!!' = **intensifier** of *most annoying*

> *(9) Throwing up at 6:00 am is* ***always*** *fun #not*
> → 'is always fun' = **evaluation** [positive polarity]
> → 'always' = **intensifier** of *is fun*

Modifiers can, but are not necessarily, syntactically close to the evaluation. When possible, however, they should be included in the annotation span of the evaluation. As shown in example 37, modifiers that are part of an evaluative expression should be included in the evaluation span. Modifiers that are not included in the evaluation span (e.g. punctuation marks, emoticons) can be linked to the evaluation they alter by means of drag and drop.

Brat howto
In brat, modifiers should be linked to the evaluative expression they alter by means of drag and drop.

Irony presence

According to our definition, verbal irony arises from a clash between two evaluation polarities. This can be illustrated with the following example:

> *(10) I really love this year's summer; weeks and weeks of awful weather.*

In sentence 10, the irony results from a polarity inversion between the literal evaluation ('I really love this year's summer'), which is positive, and the intended one ('I hate this year's summer'), which is implied by the context ('weeks and weeks of awful weather').

Irony involves a polarity clash between what is literally said and what is actually meant. What is actually meant can be explicitly mentioned, or it can be implicit.

In the former situation, the literal (ironic) evaluation is opposite to another literal evaluation in the text (e.g. 'Yay for school today! hate it...'). In the latter situation, the literal evaluation is opposite to an implied evaluation that can be inferred by common sense or world knowledge, for instance 'I appreciate you sneezing in my face'. Although 'sneezing in my face' is not an evaluation, it evokes a negative sentiment that contrasts with the literally positive statement 'I appreciate'. Section B elaborates on the annotation of implicit evaluations, or *evaluation targets*.

Annotators should carefully analyse the evaluation(s) expressed in each text and define whether the text under investigation is ironic by means of a clash or not. Additionally, a confidence score (low, medium or high) should be given for this annotation. It is possible, however, that an instance contains another form of irony: there is no polarity clash between what is said and what is meant, but the text is ironic nevertheless (e.g. descriptions of situational irony). These instances should be included in the category **other types of irony**. Instances that are not ironic should be annotated likewise. The three main annotation categories are listed below:

- **Ironic by means of a clash:** the text expresses an evaluation whose literal polarity is opposite to the intended polarity.
- **Other type of irony:** there is no clash between the literal and the intended evaluation, but the text is still ironic.
- **Not ironic:** the text is not ironic.

Sentences 11 and 12 are examples of ironic texts in which the literally expressed evaluation is opposite to the intended one. In example 11, the irony results from a clash between the literally positive 'Yay can't wait!' and 'Exams start tomorrow', which implicitly conveys a negative sentiment. In contrast to sentence 11, the irony in example 12 can only be understood by the presence of the hashtag *#not*. Without this hashtag, it is not possible to perceive a clash between what is literally said and what is implied (i.e. 'my little brother is not awesome'). For similar cases, annotators indicate that a hashtag is required to understand the irony.

(11) Exams start tomorrow. Yay, can't wait! #sarcasm
→ the message is **ironic by means of a clash**: the polarity of the literally expressed evaluation 'Yay, can't wait!' is positive, whereas the intended evaluation is negative (having exams is generally experienced as unpleasant).

> *(12) My little brother is absolutely awesome! #not.*
> → the message is **ironic by means of a clash**: the polarity of the literal evaluation 'is absolutely awesome!' is positive, whereas the intended evaluation is negative.

Brat howto

In brat, if an irony-related hashtag (i.e. '#sarcasm', '#irony' or '#not') is required to understand that the text is ironic by means of a clash, annotators should check the tick box **'hashtag indication needed'**.

Instances that are ironic but not by means of a clash, should be annotated as **other types of irony**. Sentences 13 to 15 are examples that belong to this category. Sentences 14 and 15 present descriptions of situational irony.

> *(13) "@Buchinator_ : Be sure you get in all those sunset instagrams before the sun explodes in 4.5 billion years." Look at your next tweet #irony*

> *(14) Just saw a non-smoking sign in the lobby of a tobacco company #irony*

> *(15) My little sister ran away from me throwing a water balloon at her and fell into the pool... #irony.*

Examples of **non-ironic** messages are presented in examples 16 to 19. As non-ironic, we consider instances that do not contain any indication of irony (example 16) or instances that contain insufficient context to understand the irony (example 17). Additionally, the category encompasses tweets in which an irony-related hashtag is used in a self-referential meta-sentence (example 18), or functions as a negator (example 19).

> *(16) Drinking a cup of tea in the morning sun, lovely!*

> *(17) @GulfNewsTabloid Wonder why she decided to cover her head though! #Irony*

> *(18) @TheSunNewspaper Missed off the #irony hashtag?*

(19) *Those that are #Not #BritishRoyalty should Not presume #Titles or do any #PublicDuties*

Brat howto

Whether an instance i) is ironic by means of a clash, ii) contains another type of irony or iii) is not ironic at all, should be annotated on the dummy token ¶ preceding each text. The category ***other type of irony*** is separated into instances describing situational irony (category ***situational irony***) and instances expressing other forms of verbal irony (category ***other***).

Irony harshness

Sarcasm is sometimes considered a bitter or sharp form of irony that is meant to ridicule or hurt a specific target (Attardo 2000, Barbieri and Saggion 2014, Lee and Katz 1998b). If a tweet is considered ironic by means of a clash, annotators should indicate the **harshness** of the expressed evaluation (i.e., whether the irony is used to ridicule or hurt a person/a company,...). This can be done on a **two-point scale from 0 to 1**, where 0 means that the evaluation is not harsh and 1 that the evaluation is harsh. Additionally, a confidence score (low, medium or high) should be given for this annotation.

(20) *Well this exam tomorrow is gonna be a bunch of laughs #not*
→ the message is ironic by means of a clash: the polarity of the literal evaluation is opposite to that of the intended evaluation
→ the ironic evaluation is **not harsh** (score 0)

(21) *Yeah you sure have great communication skills #not*
→ the message is ironic by means of a clash
→ the ironic evaluation is **harsh** (score 1), the evaluation is aimed at a person and is ridiculing

Practical remark

For convenience and to speed up the annotation, a harshness score of 0 need not be annotated explicitly. When there is no harshness score indicated, the message is considered not harsh.

Evaluation target

As mentioned in Section B, irony often tends to be realised implicitly (Burgers 2010). This means that one of the opposite evaluations may be expressed in an **implicit** way; its polarity has to be inferred from the context or by world knowledge/common sense. Such text spans are referred to as the **evaluation target**; their implicit sentiment contrasts with the literal evaluation. In brat, targets should always be linked to the evaluative expression(s) they contrasts with.

Like evaluative expressions, the implicit polarity of an evaluation target can be **positive, negative** or **unknown**. It can also be **neutral** when the target corefers to another (the actual) target. In example 22 for instance, 'you' is a neutral target that refers to '7 a.m. bedtimes', whose implicit polarity is negative given the context. There are no restrictions as to what forms evaluation targets can take: they can be expressed by a complement to a verb phrase (i.e., verb + verb, verb + adverb, verb + noun) (example 23), or by a noun phrase (e.g. 'Christmas Day', 'school'), etc. Two targets that are connected by a conjunction should be annotated separately (example 24).

> *(22) Ahh 7 a.m. bedtimes, how I've missed you #not #examproblems*
> → ''ve missed = **evaluation** [positive polarity]
> → 'you = **target** of ''ve missed', which refers to the actual target **'7 a.m. bedtimes'**

> *(23) I did so well on my history test that I got an F-!*
> → 'did so well' = **evaluation** [positive polarity]
> → 'got an F-' = **target** of 'did so well'

> *(24) I just love when the dog of the neighbours barks unstoppably and I can't sleep #not*
> → 'just love' = **evaluation** [positive polarity]
> → 'just' = **intensifier** of 'love'
> → 'the dog of the neighbours barks unstoppably' = **target** of 'just love'
> → 'can't sleep' = **target** of 'just love'

Brat howto
In brat, evaluation targets should always be linked to the evaluation they contrasts with by means of drag and drop. They cannot cross sentence boundaries.

A coreferential relation between two evaluation targets can also be added by means of drag and drop.

Embedded evaluations

Sometimes, an evaluation is contained by another evaluation (e.g. sentence 25). This is called an *embedded* evaluation and needs to be annotated as well. Similarly to evaluative expressions, the polarity of embedded evaluations can be **positive, negative** (or **unknown** in the case there is not sufficient context), and its prior polarity may be changed by modifiers.

> *(25) I'm really looking forward to the awful stormy weather that's coming this week.*
> → 'i'm really looking forward to' = **evaluation** [positive polarity]
> → 'really' = **intensifier** of 'looking forward to'
> → 'the awful stormy weather that's coming this week' = **target** of 'really looking forward to'
> → 'awful' = **(embedded) evaluation** [negative polarity]

Annotation procedure

In what follows, we present the different steps in the annotation procedure. It should be noted that, even if a message is not ironic or contains another type of irony than the one based on a polarity clash, annotators should annotate all evaluations that are expressed in the text under investigation. We refer to Section B for detailed annotation examples in brat.

1. **Based on the definition, indicate for each text whether it: i) is ironic by means of a clash, ii) contains another type of irony or iii) is not ironic** and indicate a confidence score for this annotation.

   - **Ironic by means of a clash**: the text expresses an evaluation whose literal polarity is the opposite of the intended polarity.
   - **Other type of irony:** there is no contrast between the literal and the intended evaluation, however, the text is still ironic (e.g. descriptions of situational irony).
   - **Not ironic:** the text is not ironic.

2. **If the text is ironic by means of a clash:**

   - In the case of tweets, indicate whether an **irony-related hashtag** (*#sarcasm*, *#irony*, *#not*) is required to recognise the irony.
   - Indicate the **harshness** of the irony on a two-point scale (0-1) and indicate a confidence score for this annotation.

3. **Annotate all evaluations contained by the text**

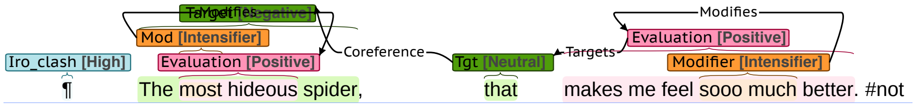
   - Indicate the polarity of each evaluation.
   - If present, annotate modifiers and link them to the corresponding evaluation.
   - If present, annotate the evaluation target(s) and link it/them to the evaluation it is in contrast with.
     * If the target refers to another target, link them by means of a coreferential relation.
     * Indicate the implicit polarity of the target based on context, world knowledge or common sense.

4. **If present, annotate embedded evaluations.**

5. **Proceed with the next text.**

Brat annotation examples

(26)

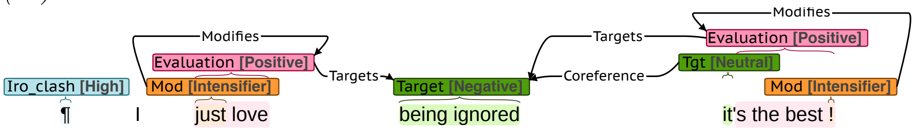


The most hideous spider, that makes me feel sooo much better. #not

- the message is ironic by means of a clash
  → the irony is not harsh
- 'makes me feel so much better' = **evaluation** [positive polarity]
- 'sooo much' = **intensifier** of 'makes me feel better'
- 'that' = **target** that refers to 'the most hideous spider'
- 'the most hideous spider' = **target** of 'makes me feel sooo much better' [implicit polarity = negative]

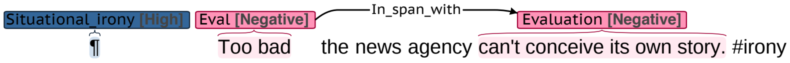
- ‘most hideous’ = **(embedded) evaluation** [negative polarity]
- ‘most’ = **intensifier** of ‘hideous’

(27)



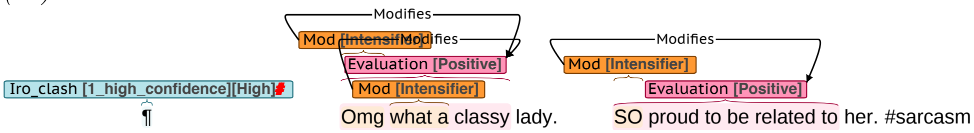
- the text is ironic by means of a clash
  → the irony is not harsh
- ‘just love’ = **evaluation** [positive polarity]
- ‘just’ = **intensifier** of ‘love’
- ‘it’ = **target** that refers to ‘being ignored’
- ‘being ignored’ = **target** of ‘just love’ [implicit polarity = negative]
- ‘’s the best!’ = **evaluation** [positive polarity]
- ‘!’ = **intensifier** of ‘’s the best’

(28)



- the text contains another type of irony, it describes an ironic situation
- ‘Too bad ... can’t conceive its own story’ = **evaluation** [negative polarity]

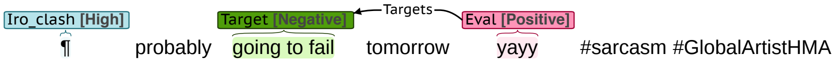
(29)



- the text is ironic by means of a clash
  → the irony is harsh

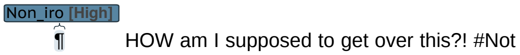
- ‘Omg what a classy’ = **evaluation** [positive polarity]
- ‘Omg’ = **intensifier** of ‘what a classy’
- ‘what a’ = **intensifier** of ‘classy’
- ‘SO proud to be related to’ = **evaluation** [positive polarity]
- ‘SO’ = **intensifier** of ‘proud to be related to’

(30)



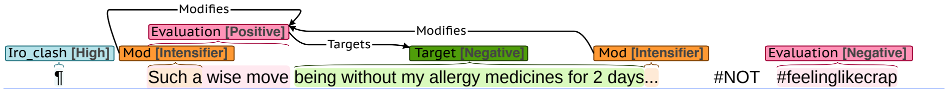
- the text is ironic by means of a clash
  → the irony is not harsh
- ‘yayy’ = **evaluation** [positive polarity]
- ‘going to fail’ = **target** of ‘yayy’ [implicit polarity = negative]

(31)



- the text is not ironic

(32)

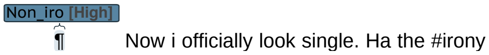


- the text is ironic by means of a clash
  → the irony is not harsh
- ‘Such a wise move’ = **evaluation** [positive polarity]
- ‘Such a’ = **intensifier** of ‘wise move
- ‘...’ = **intensifier** of *Such a wise move*

- ‘being without my allergy medicines for 2 days’ = **target** of ‘Such a wise move’
- ‘#feelinglikecrap’ = **evaluation** [negative polarity]

  → Here, the irony is made obvious in two ways: i) a clash between an explicit and implicit sentiment expression (‘Such a wise move’ vs. ‘being without my allergy medicines for 2 days’), and ii) a clash between two explicit sentiment expressions (‘Such a wise move’ vs. ‘#feelinglikecrap’).

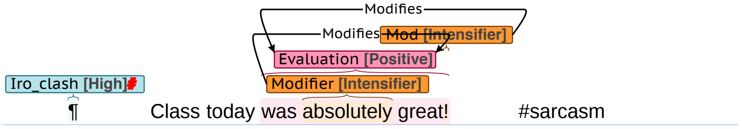
(33)



Now i officially look single. Ha the #irony

- the text is not ironic.

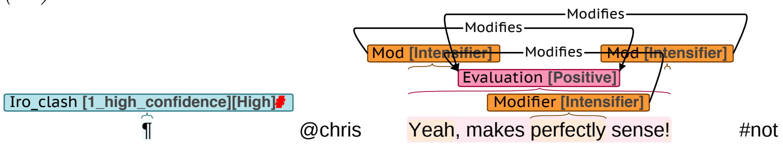
(34)



Class today was absolutely great! #sarcasm

- the text is ironic by means of a clash
  → the irony is not harsh
  → the hashtag ‘#sarcasm’ is required to understand the irony
- ‘was absolutely great!’ = **evaluation** [positive polarity]
- ‘absolutely’ = **intensifier** of ‘was ... great!’
- ‘!’ = **intensifier** of ‘was absolutely great’

(35)

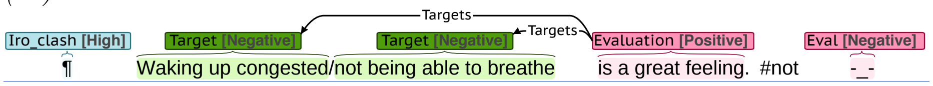


@chris Yeah, makes perfectly sense! #not

- the text is ironic by means of a clash
  → the irony is harsh
  → the hashtag ‘#sarcasm’ is required to understand the irony

- 'Yeah, makes perfectly sense!' = **evaluation** [positive polarity]
- 'Yeah' = **intensifier** of 'makes perfectly sense!'
- 'perfectly' = **intensifier** of 'makes ... sense!'
- '!' = **intensifier** of 'makes perfectly sense'

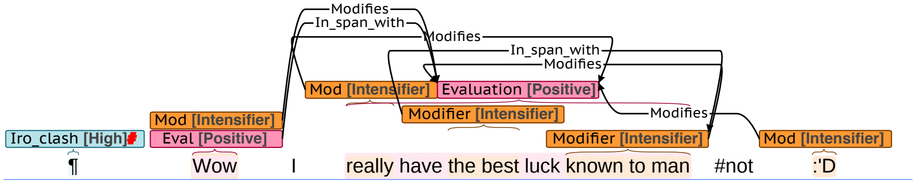
(36)



¶ Waking up congested/not being able to breathe is a great feeling. #not -_-

- the text is ironic by means of a clash
  → the irony is not harsh
- 'is a great feeling' = **evaluation** [positive polarity]
- 'Waking up congested' = **target** of 'is a great feeling' [implicit polarity = negative]
- 'not being able to breathe' = **target** of 'is a great feeling' [implicit polarity = negative]
- '-_-' = **evaluation** [negative polarity]

(37)



¶ Wow I really have the best luck known to man #not :'D

- the text is ironic by means of a clash
  → the irony is not harsh
  → the hashtag '#not' is required to understand the irony
- 'really have the best luck known to man' = **evaluation** [positive polarity]
- 'Wow' = **intensifier** of 'really have the best luck known to man'
- 'really' = **intensifier** of 'Wow ... have the best luck known to man'

- ‘the best ... known to man’ = **intensifier** of ‘Wow ... really have luck’
- ‘:’D’ = **intensifier** of ‘Wow ... really have the best luck known to man’