Lang Resources & Evaluation
https://doi.org/10.1007/s10579-018-9414-2

ORIGINAL PAPER

Exploring the fine-grained analysis and automatic detection of irony on Twitter

Cynthia Van Hee[1] · **Els Lefever**[1] · **Véronique Hoste**[1]

Abstract To push the state of the art in text mining applications, research in natural language processing has increasingly been investigating automatic irony detection, but manually annotated irony corpora are scarce. We present the construction of a manually annotated irony corpus based on a fine-grained annotation scheme that allows for identification of different types of irony. We conduct a series of binary classification experiments for automatic irony recognition using a support vector machine (SVM) that exploits a varied feature set and compare this method to a deep learning approach that is based on an LSTM network and (pre-trained) word embeddings. Evaluation on a held-out corpus shows that the SVM model outperforms the neural network approach and benefits from combining lexical, semantic and syntactic information sources. A qualitative analysis of the classification output reveals that the classifier performance may be further enhanced by integrating implicit sentiment information and context- and user-based features.

Keywords Verbal irony · Social media · Automatic irony detection · Machine learning

✉ Cynthia Van Hee
cynthia.vanhee@ugent.be

Els Lefever
els.lefever@ugent.be

Véronique Hoste
veronique.hoste@ugent.be

[1] LT3, Language and Translation Technology Team, Department of Translation, Interpreting and Communication, Ghent University, Groot-Brittanniëlaan 45, 9000 Ghent, Belgium

Published online: 26 February 2018

1 Introduction

Irony has always played an important role in human communication, although its functions may vary: it can be the instrument of a moral lesson (i.e. 'Socratic irony') (Vlastos 1987), ridicule or scorn (Wilson and Sperber 1992), a face-protecting strategy when expressing criticism (Brown and Levinson 1987), or a way to express creativity in writing (Veale and Hao 2009). Understanding irony is therefore crucial if we want to improve our understanding of human language and communication.

As a result of the digital (r)evolution, more and more communication takes place online, which has stimulated text mining and more concretely sentiment analysis research. Sentiment analysis, which aims to automatically extract positive and negative opinions from online text, has become one of the main research domains in natural language processing. State-of-the-art sentiment classifiers have been developed in the context of specialised shared tasks like SemEval (Nakov et al. 2016) and have flourished in industry through commercial applications (Liu 2012). However, many applications struggle to maintain high performance when applied to ironic text (Maynard et al. 2014; Ghosh and Veale 2016). The following examples illustrate this problem.

(1) *I love how my mom says she can count on Rion more than me. #not #jealous* (Example taken from the SemEval-2015 Task 11 data by Ghosh et al. (2015)).

Regular sentiment analysis systems would probably classify example 1 as positive, whereas the intended sentiment is undeniably negative. In this tweet, the irony is indicated with the hashtag *#not*, but many other ironic instances 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 example 2 does not feel blessed at all and wants to communicate the opposite. This can be inferred from the clash between the positive sentiment statement "I feel so blessed" and the negative sentiment associated with getting ocular migraines.

To enhance the performance of sentiment analysis, and even "any model that hopes to make sense of human communication or expression" (Wallace 2015, p. 468), it is crucial to build computational models capable of detecting irony. To achieve this, it is key to understand how irony is linguistically realised and to identify aspects and forms of irony that are susceptible to computational analysis.

State-of-the-art approaches to irony detection often utilise hashtags (e.g. *#irony, #sarcasm, #not*) assigned by the author of the text to label instances in an irony corpus, but this has shown to introduce noise into the labelled training data (Kunneman et al. 2015). For the current study, we also collected ironic tweets using the above hashtags, but supplemented them with manual annotations using a fine-grained annotation scheme (Van Hee et al. 2016b). Based on the annotations, we examine how different kinds of irony are realised in our corpus and we translate this information into a rich feature set to detect irony automatically. As part of our comprehensive approach, we explore the feasibility of automatic irony detection using a support vector machine (SVM) and a deep learning approach, the latter of which has recently gained considerable popularity for text modelling tasks.

The remainder of this paper is structured as follows. Section 2 discusses related research on defining and modelling irony. Section 3 introduces the corpus and zooms in on the fine-grained annotation scheme for irony. The experimental setup and results discussion are the topic of Sect. 4. Finally, Sect. 5 concludes the paper and presents some perspectives for future research.

2 Irony research

Defining irony is an arduous task, and various conceptualisations and theories have been established in the past. According to Kreuz and Roberts (1993), four types of irony can be distinguished: (1) Socratic irony and (2) 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), (3) irony of fate, which involves an incongruency between two situations, and (4) verbal irony, which implies a speaker who intentionally says the opposite of what he believes. However, theorists traditionally distinguish between **situational** and **verbal** irony. Situational irony would include dramatic irony and irony of fate as described by Kreuz and Roberts (1993) and refers to situations that fail to meet some expectations (Shelley 2001) (see example 3).

(3) “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).

As explained by Burgers (2010), the classical definition of verbal irony is attributed to the author Quintilian (1959) and states that verbal irony implies saying the opposite of what is meant. Until today, this approach has influenced many conceptualisations of irony, one of the most well-known probably being Grice’s theory of conversational implicature (1975, 1978). Although it has faced criticism in the past, this theory is often referred to in linguistic and computational approaches to irony.

2.1 Conceptualisations of verbal irony

In what follows, we highlight seminal work in irony literature and describe the state of the art in automatic irony detection. We discuss the most relevant studies for the present research, but refer to the papers by Wallace (2015) and Joshi et al. (2017) for a more detailed overview. Important to note is that when discussing related research, we refer to irony using the terminology employed by the corresponding researchers [i.e. ‘sarcasm’ or ‘(verbal) irony’]. We start with the seminal work by Grice, who posits that irony is a flouting of the conversational maxim of Quality (i.e. ‘do not say what you believe to be false’) to make clear that what he means differs from what he literally says, and hereby expresses a feeling, attitude or evaluation (Grice 1978). Although Grice’s (1975) theory of conversation has widely impacted language philosophy and semantics, his view on verbal irony has been questioned (e.g. Sperber and Wilson 1981; Giora 1995). In what follows, we briefly discuss some critiques towards and alternatives to his approach.

According to Sperber and Wilson (1981), Grice’s theory fails to explain the purpose of irony and does not cover more subtle variants of irony, including understatements and allusions (examples 4 and 5).

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

(5) (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).

As an alternative, they propose the **Echoic Mention Theory**, stating that ironic utterances implicitly allude to a previous proposition, and thereby express the speaker’s negative attitude towards it. As such, the irony in examples 4 and 5 targets the speaker’s negative attitude towards the hearer’s previously uttered claim that the customer is upset and that London is a fantastic city.

Another post-Gricean approach to verbal irony that is worth mentioning is the **Pretense Theory** by, among others, Clark and Gerrig (1984), Currie (2006) and Kumon-Nakamura et al. (1995). In accordance with Sperber and Wilson (1981), irony is considered allusive, but it does not necessarily allude to a previous proposition, it can also refer to some failed expectation or norm. The irony involves pragmatic insincerity, such as insincere compliments (e.g. “You sure know a lot”), rhetorical 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]

Giora (1995) describes irony as an **indirect negation** strategy, which seems to reconcile elements from both the traditional or Gricean approach and the so-called ‘post-Gricean’ theories 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. In the why of using irony, Giora (1995) sees a politeness strategy enabling its users to negate or criticise something in a face-protecting way.

Many theories of verbal irony have been established, but all of them appear to have some elements in common, such as an opposition between what is said and what is intended and the fact that irony involves an evaluation of something or someone (Burgers 2010; Camp 2012). And while Grice’s (1975) theory has been criticised from various points of view (e.g. see earlier), we believe that his approach covers a substantial number of ironic instances. In fact, the main criticism towards

[1] The above examples are taken from Kumon-Nakamura et al. (1995).

his theory is that it fails to explain (1) more subtle variants of irony, and (2) why irony would be preferred over a sincere utterance. However, 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) and identify linguistic devices to realise irony rather than different forms of irony (e.g. hyperboles and rhetorical questions can be a way to express the opposite of what one intends to say, but they are not necessarily different types of irony). Consequently, like that of Burgers (2010), our working definition of irony is based on the traditional approach (see Sect. 3.1).

It is noteworthy that we will use the term 'irony' throughout this paper and do not distinguish between irony and sarcasm, since opinion on the difference between the two is still very much divided. While some theorists consider sarcasm and irony the same, others posit that they do differ in some respects, claiming that sarcasm is a form of verbal irony that has a more aggressive or ridiculing tone (Attardo 2000), is directed at a person (Sperber and Wilson 1981) and is used intentionally (Gibbs et al. 1995), while irony is not. Given that many of these features are often difficult to recognise, it is unclear, however, whether they provide sufficient evidence of a clear-cut distinction between irony and sarcasm.

2.2 Computational approaches to irony

Research in natural language processing (NLP) has recently seen various attempts to tackle automatic irony detection. As described by Joshi et al. (2017), computational approaches to irony can be roughly classified into rule-based and (supervised and unsupervised) machine learning-based. While rule-based approaches mostly rely upon lexicon and word-based information, machine learning often exploits rich feature sets that are either manually defined or learned in neural networks.

Early work by Davidov et al. (2010) describes a semi-supervised approach to irony detection exploiting punctuation and syntactic patterns as features. Their system was trained on Amazon and Twitter data and obtained F-scores of respectively 79% and 83%. Similarly, Bouazizi and Ohtsuki (2016) extracted more than 300,000 part-of-speech (PoS) patterns and combined them with lexical (e.g. bags of words), sentiment (e.g. positive and negative sentiment word values) and other syntactic features (e.g. number of interjections), which yielded an F_1-score of 81%. González-Ibáñez et al. (2011) combined standard lexical features with pragmatic information such as frowning emoji and @-replies. They used sequential minimal optimisation (SMO) and logistic regression as classifiers and obtained an accuracy of 65% in a non-balanced dataset (33% ironic, 67% non-ironic tweets). Reyes et al. (2013) defined features based on conceptual descriptions in irony literature, being *signatures, unexpectedness, style* and *emotional scenarios* and experimented with Naïve Bayes and decision trees. They distinguished tweets with the *#irony* hashtag from tweets with the hashtags *#education, #humor* or *#politics* and found that the best scores were obtained by distinguishing *#irony* from *#humor* tweets. Kunneman et al. (2015) pioneered irony detection in Dutch tweets using word *n*-gram features and a Balanced Winnow classifier. They trained a classifier by contrasting the hashtag-labelled irony tweets (i.e. tweets with a

#sarcasm hashtag were considered ironic, tweets devoid of such a hashtag were not ironic) against a background corpus without such hashtags, which yielded an AUC-score of 0.85. Manual inspection of the tweets classified as sarcastic in the background corpus revealed, however, that only 35% were effectively sarcastic, showing that sarcasm detection is an arduous task in an open setting. Van Hee et al. (2016a) introduced a fine-grained annotation scheme for irony detection for both English and Dutch ironic tweets and were the first to apply such a fine-grained irony annotation to Dutch tweets. The researchers combined lexical with sentiment, syntactic and semantic Word2Vec cluster features for irony detection in English tweets using an SVM and obtained a top F_1-score of 68%. Similarly, Joshi et al. (2016) used an SVM classifier and expanded their set of lexical and sentiment features with different word embedding features. They showed that incorporating Word2Vec and dependency weight-based word embeddings results the most beneficial for irony detection, yielding F-scores of up to 81%. Their dataset consisted of book reviews tagged with a ‘sarcasm’ (i.e. positive class) and a ‘philosophy’ (i.e. negative class) label.

Riloff et al. (2013) demonstrated that irony mostly involves a positive evaluation (e.g. ‘cannot wait’) targeting a negative situation (e.g. ‘go to the dentist’). They took a bootstrapping approach to learn negative situation phrases in the vicinity of positive seed words like ‘love’ to detect ironic utterances. Combining this contrast method with bag-of-word features exploited by an SVM classifier yielded an F_1-score of 51%. Comparably, Joshi et al. (2015) exploited explicit and implicit incongruity features, which outperformed lexical features on a corpus of hashtag-labelled tweets (F_1 = 89%), but not on a manually-labelled irony corpus (F_1 = 61%). Karoui et al. (2015) were one of the first to approach irony detection by identifying factual oppositions in tweets. Observing that French ironic tweets often contain negations, the researchers integrated pragmatic context by checking facts in potentially ironic tweets.

More recent work has approached irony detection using deep learning, which makes use of neural networks based on continuous automatic features instead of manually defined ones. Amir et al. (2016) employed convolutional neural networks (CNNs) to automatically learn irony indicators based on content and user embeddings. They revealed that contextual features significantly improve irony detection performance and reported accuracies of up to 87%. Their deep learning approach slightly outperformed that of Bamman and Smith (2015), where contextual features for irony detection were exploited using a logistic regression algorithm. Poria et al. (2016) approached irony detection using a CNN exploiting baseline features (i.e. inherent semantics deduced by the network) and features obtained through pre-trained CNNs for sentiment, emotion and personality classification. The classifier obtained an F_1-score of 87% with baseline features and 91% with pre-trained embeddings. Ghosh and Veale (2016) compared the performance of an SVM model (F = 73%) exploiting bag-of-word features, PoS information and sentiment features to a neural network model learning word embeddings. They demonstrated that the latter architecture [CNN + LSTM + DNN (deep neural network)], outperformed the SVM model, yielding an F-score of 92% when irony hashtags were included in the data.

A substantial part of the above studies approach irony detection using text-based features, but there is a growing tendency towards exploiting extra-textual features related to the author or context of a tweet. Among other researchers Amir et al. (2016) and Bamman and Smith (2015) demonstrated the benefits of including previous tweets as features, author profile information, the number of interactions between two users, and so on.

While most approaches have focused on English data, irony detection has also been investigated in other languages, including Italian, French, Czech, Portuguese, Greek, Indonesian and Dutch. It can be observed that a number of features are the same across languages, such as bags of words, PoS patterns and sentiment features. However, some features do capture language-specific characteristics of irony, such as the frequent use of negations in French (Karoui et al. 2015), variations in verb morphology and *cross-constructions* in Portuguese (Carvalho et al. 2009), political discourse in Italian (Barbieri et al. 2014), words that showcase figurative language in Greek (Charalampakis et al. 2016) and words whose polarity varies depending on its context in Indonesian (e.g. the Indonesian word for ‘student’ means ‘low price’ in particular contexts) (Lunando and Purwarianti 2015).

It is important to note that many of the discussed papers in this section make use of very large training corpora (up to 812K tweets), whereas the current corpus is limited to 4.7K tweets. Moreover, in the above studies, training data is often obtained by collecting tweets using hashtags like *#irony* and labelling them accordingly. An important contribution of this paper is that, after collecting data with irony hashtags, all tweets were manually labelled based on a fine-grained annotation scheme for irony (Van Hee et al. 2016b).

3 Constructing a corpus of ironic tweets

In this section, we describe the construction of an English irony corpus and the development of fine-grained annotation guidelines for irony (Van Hee et al. 2016b) as introduced in Van Hee et al. (2016a). An inter-annotator experiment was set up to assess the validity of the guidelines.

3.1 Data collection and annotation

To operationalise the task of irony detection, we constructed a dataset of 3000 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[2], a Python library to access the Twitter API[3] providing programmatic access to read Twitter data.

To minimise noise in the dataset, all tweets were manually labelled. Given the absence of fine-grained coding principles for this task, we developed a new annotation scheme that is described in Sect. 3.2. Cleaning of the corpus involved the

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

[3] https://dev.twitter.com/rest/public.

removal of duplicates, retweets, and non-English tweets, as well as handling non-ascii and html-escaped characters. To facilitate annotation of the data using Brat (Stenetorp et al. 2012), emoji were converted to UTF8 descriptions using the Python `emoji` module.[4]

While a number of annotation schemes for irony have been developed recently (e.g. Riloff et al. 2013; Bosco et al. 2016; Stranisci et al. 2016), most of them describe a binary distinction (i.e. ironic vs. not-ironic) or flag irony as part of sentiment annotations. By contrast, Karoui et al. (2017) defined explicit and implicit irony activations based on incongruity in ironic tweets and defined eight fine-grained categories of pragmatic devices that realise such an incongruity, including analogy, hyperbole, rhetorical question, oxymoron, etc. While the typology provides valuable insights into the linguistic realisation of irony, the inter-annotator agreement study demonstrated the complexity of identifying such pragmatic devices, hence it is not clear to which extent such the distinction would be computationally feasible.

In the following section, we describe the construction of a fine-grained annotation scheme for irony in social media text and report the results of an inter-annotator agreement study to assess the validity of the guidelines. The scheme is, to our knowledge, the first to allow annotators to distinguish between different types of irony. Moreover, it allows them to indicate, below sentence level, specific text spans in order to grasp the mechanisms of irony in text.

3.2 Annotation guidelines

Literature shows that irony is often realised by means of a polarity contrast (cf. Sect. 2.1). As the starting point of the annotation process, we therefore defined 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. 2016b). Such evaluations can be explicit or implicit (i.e. irony *targets* or text spans that contain no subjective words, but implicitly convey a positive or negative sentiment).

Besides ironic instances showing such a polarity contrast, the scheme allows annotators to indicate other types of irony and non-ironic instances. The three main annotation categories are listed below.

- *Ironic by means of a clash* in accordance with our definition, the text expresses an evaluation whose literal polarity is opposite to the intended polarity. The intended polarity can be explicit (e.g. “Yay for school today! #hate it”), or implicit (e.g. “I appreciate you sneezing in my face”).
- *Other type of irony* there is no contrast between the literal and the intended evaluation, but the text is still ironic. This category further distinguishes between instances describing situational irony and other forms of verbal irony.
- *Not ironic* the text is not ironic.

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

In case of irony resulting from a polarity clash, the annotators made two supplementary annotations to gain insights into the linguistic realisation of this type of irony. Firstly, they indicated the harshness of an instance (0 or 1), 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 (see earlier). Example 6 presents such a harsh tweet.

(6) 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 indicated whether an irony-related hashtag was required to recognise the irony, as is the case in example 7. As opposed to example 6, the tweet is not considered harsh.

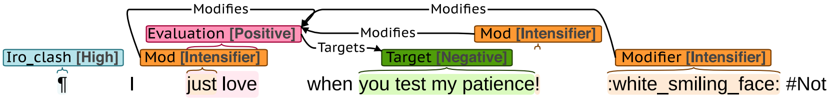
(7) This should be fun next spring. #not

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 the 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 evaluation.
- *Modifiers* (if present) words that alter the prior polarity of the evaluation (e.g. 'unbelievably thoughtful').
- *Targets* text spans whose implicit sentiment contrasts with that of the literal evaluation.

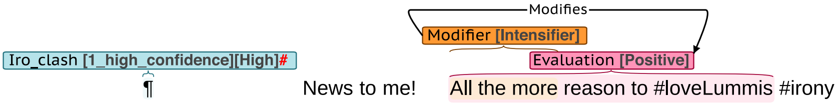
All annotation steps were done using brat, a web-based annotation tool (Stenetorp et al. 2012), some visualisations of which are shown in examples 8–12.

(8)



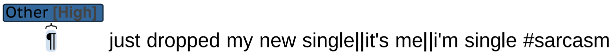
¶ I just love when you test my patience! :white_smiling_face: #Not

(9)

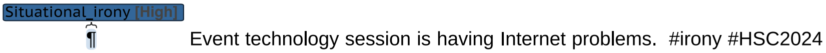


¶ News to me! All the more reason to #loveLummis #irony

(10)



¶ just dropped my new single||it's me||i'm single #sarcasm

(11) 

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

(12)

stop subtweeting :winking_face: #irony

Examples 8 and 9 illustrate irony by means of a clash. Sentence 8 contains a polarity clash between the literal evaluation ("just love") and its target ("you test my patience"), which has been assigned a negative connotation. Tweet 9 is also ironic with a polarity contrast, but unlike the previous example, the irony cannot be understood from the main text. In this case, the hashtag *#not* is required, otherwise the evaluation might as well be genuine. Examples 10 and 11 illustrate other verbal irony and situational irony, respectively. Finally, tweet 12 is not ironic, despite the presence of the hashtag *#irony*. Confidence scores (viz. low, medium or high) were added to each annotated tweet to indicate the annotator's certainty about the annotation. Whenever 'low' or 'medium' was indicated for an instance, its annotation received an additional check by one of the experts.

3.3 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. To assess the reliability of the annotations, and whether the guidelines allowed the annotators to carry out the task consistently, an inter-annotator agreement study was carried out on 100 instances from the corpus.

As metric, we used Fleiss' kappa (Fleiss 1971), a widespread statistical measure in the field of computational linguistics for assessing agreement between two or more annotators on categorical ratings (Carletta 1996). The measure calculates the degree of agreement in classification over the agreement which would be expected by chance.

Table 1 presents the inter-annotator agreement for different steps in the annotation process, including the irony type annotation, whether an irony hashtag is required to understand the irony, the level of harshness in case the tweet is ironic, and the polarity contrast annotation. 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[5], indicating that the scheme allows for a reliable annotation.

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

Table 1 Inter-annotator agreement (Fleiss' Kappa)

Annotation	Kappa κ
Ironic by clash/other/not ironic	0.72
Hashtag indication	0.69
Harshness	0.31
Polarity contrast	0.66

Table 2 Statistics of the annotated corpus: number of instances per annotation category

Ironic by means of a clash	Other type of irony		Not ironic	Total
	Situational irony	Other verbal irony		
1728	401	267	604	3000

3.4 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, 3000 English tweets with the hashtags *#irony, #sarcasm and #not* were annotated based on our fine-grained guidelines.

Table 2 presents some annotation statistics. As can be inferred from the table, most ironic instances 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* constitutes the majority of this annotation class, as compared to *other verbal irony*. Out of the total of 3000 tweets, no less than 604 were considered not ironic. Importantly, this would mean that an irony corpus based on hashtag information as gold labels can contain about 20% noise. We see several explanations for this noise. First, analysis of the data reveals that more than half of the non-ironic tweets contain the hashtag *#not*, which is often used as a negation word. Second, manual analysis showed that irony-related hashtags were sometimes used meta-linguistically (i.e. to refer to the phenomenon itself).

We further found that 72% of the ironic tweets were realised by a polarity contrast, while the remaining 30% consisted of situational irony and other verbal irony. Interestingly, almost half of the polarity contrast tweets required an irony-related hashtag to recognise the irony (cf. example 9). Hence, annotators deemed it impossible to recognise the irony without such a hashtag. Moreover, 34% of the tweets were considered harsh, meaning that the text was meant to ridicule or hurt someone. This is an interesting observation, given that irony literature states that harshness or ridicule could be distinguishing factors between irony and sarcasm, the latter of which is often considered the acrimonious form of the two (Attardo 2000). Analysis of the harshness annotation revealed a stronger correlation between harshness and the presence of the *#sarcasm* hashtag in the corpus compared to

#irony and *#not*. Also, it was observed that harsh tweets contained more second-person pronouns (1.18% of all tokens) than non-harsh tweets (0.59%).

As shown by among others Kunneman et al. (2015) and Riloff et al. (2013), ironic utterances are likely to contain markers such as interjections (e.g. 'yeah right') or intensifiers and diminishers to express hyperbole and understatements. We observed that 40% of the ironic instances in the corpus contained an intensifier (e.g. 'sooo'), while only 8% contained a diminisher (e.g. 'kinda'). 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. These observations seem to corroborate that modifiers like intensifiers and interjections are indeed often used to mark irony in tweets.

4 Automatic irony detection

To recapitulate, our corpus comprises 3000 manually annotated tweets (i.e. the 'hashtag corpus'). About 20% of them were considered non-ironic and were added to the negative class, which leaves 2396 ironic and 604 non-ironic tweets. To balance the class distribution, we expanded the latter with 1792 non-ironic tweets from a background corpus, leaving the experimental corpus with a total number of 4792 tweets. Next, the corpus was randomly split into a training and test set of respectively 80 and 20% showing a balanced class distribution. While the former was used for feature engineering and classifier optimisation, the latter functioned as a held-out test set to evaluate and report classification performance. As preprocessing, all hyperlinks and @-replies were normalised to 'http://someurl' and '@someuser' and common abbreviations were replaced by their full form[6]. Furthermore, superfluous white spaces were removed, as well as vertical bars or *pipes*. Other preprocessing steps include tokenisation and PoS-tagging (Gimpel et al. 2011), lemmatisation (Kauter et al. 2013) and named entity recognition (Ritter et al. 2011).

4.1 An SVM-based approach to irony detection

First, we approached irony detection using an SVM, as the classifier has demonstrated good performance for the task (cf. Sect. 2). Prior to constructing the model, the following feature groups were defined.

4.1.1 Information sources

As **lexical** features, we included *n*-grams, which represent a tweet as a 'bag' of its words (unigrams and bigrams) and characters (trigrams and fourgrams). Other lexical features include conditional *n*-gram probabilities based on language models. Using language model probabilities is, to our knowledge, novel in irony detection.

[6] The replacements were based on an existing abbreviations dictionary: http://www.chatslang.com/terms/abbreviations.

The language models were created with KENLM (Heafield et al. 2013) based on a background corpus comprising 1,126,128 (ironic + non-ironic) tweets collected with the Twitter API. Besides bags-of-words and language model probabilities, a set of numeric and binary features were exploited, providing information about (1) character and (2) punctuation flooding, (3) the presence of punctuation, (4) capitalisation and (5) interjections, (6) hashtag frequency and (7) the hashtag-to-word ratio, (8) emoticon/emoji frequency, and (9) tweet length. Where relevant, numeric features were normalised by dividing them by the tweet length in tokens.

To incorporate **syntactic** information, we extracted part-of-speech features indicating, for each of the 25 tags used by the Twitter tagger by Gimpel et al. (2011), whether the tag occurs in a tweet and how frequently it occurs. Another feature indicates the presence of a clash between two verb tenses in a tweet[7]. For this purpose, we used the part-of-speech output by LeTs Preprocess (Van de Kauter et al. 2013), which provides verb tense information, as opposed to the Twitter tagger. Lastly, named entity (NE) features were extracted indicating the presence and frequency of NEs in each tweet.

Six **sentiment lexicon** features were implemented based on existing lexicons: AFINN (Nielsen 2011), General Inquirer (GI) (Stone et al. 1966), MPQA (Wilson et al. 2005), the NRC Emotion Lexicon (Mohammad et al. 2013), Liu's opinion lexicon (Hu and Liu 2004), Hogenboom's emoticon lexicon (Hogenboom et al. 2015), and Kralj's emoji sentiment lexicon (Kralj Novak et al. 2015). For each lexicon, we derived five numeric and one binary feature:

- 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;
- a binary feature indicating the presence of a polarity contrast (i.e. the tweet contains at least one positive and negative sentiment word).

Sentiment lexicon features were extracted in two ways: (1) by considering all tweet tokens and (2) by taking only hashtag tokens into account, after removing the hashtag (e.g. *lovely* from *#lovely*). Negation was taken into account by flipping the polarity of a sentiment word when in the proximity of a negation word.

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 are more prone to irony use than others). To verify this assumption, we utilised semantic word clusters generated from a large background corpus equal to that of the language model features (supra). 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 a tweet. An example cluster is presented below.

[7] Following the example of Reyes et al. (2013).

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

The word embeddings were generated from an English background corpus comprising approximately 1M (ironic + non-ironic) tweets. 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.

For each of the four feature groups (containing 35, 869, 105, 96 and 200 features, respectively), a binary classifier was trained and evaluated on the held-out test set. Subsequently, a series of experiments were run to evaluate the benefits of combining the feature groups.

4.2 Experimental design and results

We made use of a support vector machine as implemented in the LIBSVM library (Chang and Lin 2011), since the algorithm has been successfully combined with large feature sets and its good performance for similar tasks has been recognised (Joshi et al. 2017).

We performed binary SVM classification using the default *radial basis function* (i.e. RBF or *Gaussian*) kernel, as preliminary experiments on our dataset showed better results using RBF than with the linear kernel. 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.

As mentioned earlier, we tested the validity of different feature types for automatic irony detection, providing lexical, syntactic, sentiment and semantic information. All feature groups were tested individually and in combination to verify whether they provide complementary information to the classifier. We compared the results to **three baselines**: a random baseline, a word *n*-gram baseline (n = 1 & 2) and a character *n*-gram baseline (n = 3 & 4). It is important to note that the results are calculated on the held-out test set. In-between results obtained through cross-validation on the development set are not included due to space constraints. As the evaluation metrics, we report accuracy, precision, recall and F_1-score, the latter three of which are calculated on the positive class (i.e. ironic) instances.

Table 3 confirms the strong baseline that present *n*-gram features, given that none of the feature groups outperforms the character *n*-gram baseline. Character *n*-grams outperforming the lexical feature group might suggest that the former work better for irony detection. This seems counterintuitive, since the lexical feature group includes information which has proven its usefulness for irony detection in related work (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.

Besides the baselines, the lexical feature group scores best for the task, but it is noteworthy that the other feature groups (i.e. syntactic, sentiment and semantic) contain much less features than the lexical group. Moreover, given that these features are not directly derived from the training data, as opposed to the bag-of-words features in the lexical group, they also seem to perform well for the task. In fact, a qualitative analysis of the classifiers' output revealed that lexical features are not the holy grail to irony detection, and that each feature group has its own strength. We observed for instance that while lexical features are strong predictors of irony (especially *ironic by clash*) in short tweets and tweets containing clues of exaggeration, sentiment features often capture ironic by clash instances that are very subjective or expressive. Recall being less than 50% however, sentiment lexicon features are insufficient to capture the ironic by clash instances in the corpus, which often contain implicit sentiment. This observation is in line with the findings of Riloff et al. (2013), reporting irony detection scores between F_1 = 14% and F_1 = 47% when using sentiment lexicons features. As opposed to lexical and sentiment features, syntactic features seem better at predicting irony in (rather) long tweets and tweets containing other verbal irony. Finally, semantic features contribute most to the detection of situational irony. In the following paragraphs, we investigate the potential of combining the individual feature groups for this task to see whether they provide complementary information.

From the results in Table 3, 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%. A similar score is achieved when combining lexical with syntactic features (i.e. F_1 = 70.07%). A closer look at the system's predictions shows that most of the ironic instances that are missed by the classifier (i.e. *false negatives*) comprise realisations of *other type of verbal irony*, which is a heterogeneous category where the irony distinction between a sincere and an ironic intent is often hard to make without more context (example 13).

(13) Trying to eat crackers on the quiet floor likeee.. Maybe if I chew slower no one will notice..

Other classification errors include examples that would be recognisable if the system could rely on world knowledge informing it that an entire day at the doctor's has a negative connotation, like being thrown in at the deep end (examples 14 and 15).

(14) Spending the majority of my day in and out of the doctor has been awesome.

Table 3 Experimental results of the individual feature groups (obtained on the test set)

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

Bold values indicate the best scores per column

Table 4 Experimental results of the combined feature groups

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

Bold values indicate the best scores per column

(15) i literally love when someone throw me in at the deep end #tough #life.

To understand how the difficulty of irony detection varies depending on how it is realised, we calculated the classifier's performance for each of the subtypes. Figure 1 shows the classification performance per subtype of the class labels (i.e. ironic vs. non-ironic). The accuracies were calculated based on the output of the best-performing combined system (viz. lexical + semantic + syntactic). The bar plot shows that the system performs best in detecting ironic tweets that are realised by a polarity contrast, whereas detecting other types of verbal irony is much more

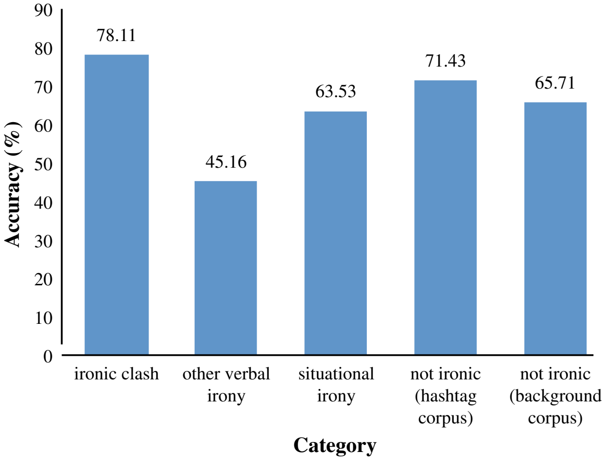


Fig. 1 Scores of the best system (lexical + semantic + syntactic features) per class label subtype

challenging. When looking at the category *not ironic*, we see that the system scores better on non-ironic tweets from the hashtag corpus as compared with non-ironic tweets from the background corpus.

Tables 3 and 4 show that generally, combining feature groups only slightly outperforms the lexical features setup. We wanted to verify whether the abundance of lexical features (as they include bags of words) influences the impact of other features when combined in an experimental setup. For this purpose, we looked at the classification performance of a system that is informed by the predictions of two different classifiers. As the lexical system performs best, we considered its output the 'system gold' and investigated whether other systems would provide supplementary information, i.e. by finding ironic tweets that the lexical system overlooks. More precisely, 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, syntactic or semantic). As such, when the lexical system predicted an instance as ironic, this was considered the final prediction for that instance, regardless of the prediction made by the other classifier. However, when the lexical system classified an instance as not ironic, whereas the other system predicted it as ironic, then the final prediction was ironic. Finally, we combined the output of all classifiers, i.e. only if all systems predicted an instance as ironic, the instance was classified as such. In all other cases, the instance was considered not ironic.

As can be deduced from Table 5, combining the output of the lexical and semantic systems yields an improvement over our best feature combination system, especially in terms of recall (R = 71.34% vs. R = 82.68%). Taking into account the predictions of all systems results in a very high precision, but at the expense of recall. This would indicate that different types of features are likely to provide complementary information, on the condition that small feature groups are not overshadowed by large ones.

Table 5 Results (%) obtained when combining the predictions of the lexical system with those of the other feature groups

Setup	Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)
lex and sent	64.72	61.87	78.97	69.38
lex and sem	**67.54**	63.85	**82.68**	**72.06**
lex and synt	67.12	63.71	81.44	71.49
lex and sent, sem, synt	60.02	**80.36**	27.84	41.35

Bold values indicate the best scores per column

4.3 A deep learning approach to irony detection

As the final step in our experimental setup, we compared the performance of our SVM approach to that of a deep learning approach based on neural networks, which have recently shown to work well for irony detection tasks (e.g. Amir et al. 2016; Poria et al. 2016). For the experiments, we adopted the standard architecture of LSTM as proposed by Hochreiter and Schmidhuber (1997). LSTM stands for *Long-Short Term Memory network* and is able to handle sequential data and capture long-term dependencies. As such, the architecture has proven especially useful for text modelling purposes, from automatic translation (e.g. McCann et al. 2017) to sentiment analysis (e.g. Ayata et al. 2017) and irony detection (e.g. Ghosh and Veale 2016). The hypothesis that polarity shifts in an utterance may indicate irony means that a model should be able to take the immediate context (i.e. surrounding words) into account. This led to the choice of using a recurrent neural network (RNN), which is able to take such context into account. LSTMs are a special kind of recurrent neural networks which have shown to outperform the latter on language modelling (Gers and Schmidhuber 2001). For the implementation of our deep learning approach, we made use of Keras (Chollet 2015), a neural networks API written in Python and capable of running on top of deep learning libraries like TensorFlow and Theano. The latter was used as backend engine for the current experiments.

As mentioned earlier, we implemented a basic LSTM architecture making use of word embeddings as features. More precisely, we built a sequential model consisting of an embeddings layer, followed by an LSTM layer and a dense layer outputting a two-dimensional output layer to pass to the activation function. We experimentally defined the architecture's parameters such as the batch size, the number of epochs and LSTM dimensions, which are detailed in the results table. We used Softmax as the activation function and Adam (Kingma and Ba 2014) for parameter optimisation, with the learning rate set to 0.001.

We trained word embeddings on our training corpus (cf. Sect. 4) using the Word2Vec (Mikolov et al. 2013) algorithm. However, as the dataset is rather small, we also experimented with pre-trained word embeddings created with the GloVe (Pennington et al. 2014) algorithm. The pre-trained word embeddings that were used in this paper are trained on Wikipedia 2014 + Gigaword 5 corpora (6B

Table 6 Results of our LSTM-based approach to irony detection

Word embeddings	Architecture parameters	Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)
(1) Irony training data (Twitter)	Batch size = 150, epochs = 5, LSTM dim. = 100	64.72	67.38	58.76	62.78
(2) Wikipedia and Gigaword	Batch size = 10, epochs = 5, LSTM dim. = 50	62.94	64.38	60.00	62.11
(3) Twitter	Batch size = 10, epochs = 5, LSTM dim. = 50	**68.27**	**68.28**	**69.69**	**68.98**

Bold values indicate the best scores per column tokens) and Twitter (27B tokens). We tested 25, 50, 100, 200 dimensional vector models, but the best results were obtained with 50-dimensional models.

Table 6 presents the results obtained with the deep learning approach using word embeddings generated from three different corpora. The table shows that the setup with Twitter-based word embeddings (setup 3) scores best, yielding an F_1-score of 68.98% and showing a good balance between precision and recall. That this setup scores better than the first two makes intuitive sense, as it relies on word embeddings that are trained on a very large Twitter corpus (27B tokens), as opposed to the embeddings generated from the training data (40K tokens) and Wikipedia (6B tokens). The latter scores slightly lower than setup 1 although the word embeddings are trained on a much larger corpus, but this is probably due to the genre difference.

The table reveals that none of the approaches outperform the SVM-based irony classifier exploiting lexical, syntactic and semantic features (F_1 = 70.11%, see Table 4). When comparing the results with the semantics feature group (i.e. word embedding features combined with an SVM classifier, see Table 3), we observe that only setup 3 outperforms the semantic features. However, all three deep learning setups score better in terms of precision.

When comparing the scores with other deep learning approaches to irony detection (e.g. Ghosh et al. 2015; Amir et al. 2016; Poria et al. 2016), we see that our results are lower, which may, on the one hand, be due to our rather small dataset, but which may also indicate the necessity to conduct more elaborated deep learning experiments on the other. For instance Poria et al. (2016) considerably outperformed (+ 20%) their SVM model using a complex neural networks architecture that combines convolutional neural networks (CNNs) with LSTMs and deep neural networks (DNNs), yielding an F_1-score of approximately 92%. It is important to note, however, that irony-related hashtags were retained in the dataset.

We believe that deep learning techniques are promising for irony detection research, given their good performance with minimal feature engineering efforts. Nevertheless, an important first step for future research would be the inclusion of context and user-based features in our SVM model. In fact, among others Amir et al. (2016) have shown that their deep learning approach using contextual features only slightly outperformed that of Bamman and Smith (2015), who used logistic regression with contextual features for irony detection.

5 Conclusions and future work

In this paper, we present a comprehensive approach to automatic irony detection. We started with assembling a new irony corpus consisting of manually annotated English tweets. Given the lack of reliable annotation guidelines for our purpose, we established a new set of coding principles that allow to identify fine-grained irony categories and mark specific text spans that realise the irony in an utterance (Van Hee et al. 2016b).

Having at hand a manually annotated irony dataset, we explored the feasibility of automatic irony detection by making use of two state-of-the-art approaches. First, we developed an SVM-based pipeline exploiting 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. The experiments revealed that, although lexical, semantic and syntactic features achieved a 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%), an observation that is in line with that of (Riloff et al. 2013). An explanation might be that the most discriminating features in the lexical feature group are ‘eclipsed’ by the large number of bag-of-word features. Combining the feature groups, however, results beneficial to the classification performance, yielding a top F_1-score of 70.11%. Combining the output of different models further enhanced classification performance by 1.5 point. Comparison with the state of the art is difficult, given that many of the discussed papers make use of much larger training corpora (up to 812K tweets), whereas the present study worked with a small dataset. Another important difference is that related research often relies on hashtag labels, whereas for this study, both training and test corpus were manually annotated. Our system compares favourably, however, to the work published by González-Ibáñez et al. (2011) and Riloff et al. (2013), which are the most comparable to the present research.

As observed in our corpus and stated by Joshi et al. (2017), irony is often realised by means of a polarity contrast, with one of the polarities often being implicit. Such implicit polarity expressions would enable us to recognise polarity contrasts in tweets like example 14, which cannot be captured using sentiment lexicons. Trying to model such implicit sentiment expressions will constitute an important research direction in the future.

Our second approach to irony detection made use of deep learning techniques based on an LSTM network. We used word embeddings derived from the training corpus and pre-trained word embeddings based on large Wikipedia en Twitter corpora and observed that the latter worked best, yielding an F_1-score of 69.89%. None of the deep learning setups improved, however, our SVM approach, which demonstrates the strength of the algorithm for the current task, especially with our (rather limited) dataset. Nevertheless, feeding the networks with other features than word embeddings, as well as experimenting with more complex networks will be important directions in future research.

Acknowledgements The work presented in this paper was carried out in the framework of the AMiCA IWT SBO-Project 120007, funded by the government Flanders Innovation and Entrepreneurship (VLAIO) agency.

References

Amir, S., Wallace, B. C., Lyu, H., Carvalho, P., & Silva, M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. CoRR abs/1607.00976

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

Ayata, D., Saraclar, M., Ozgur, A. (2017). BUSEM at SemEval-2017 task 4: A sentiment analysis with word embedding and long short term memory RNN approaches. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval'17)* (pp. 777–783). ACL, Vancouver, Canada.

Bamman, D., Smith, N. A. (2015). Contextualized sarcasm detection on Twitter. In *Proceedings of the ninth international conference on web and social media (ICWSM'15)* (pp. 574–577). AAAI, Oxford, UK.

Barbieri, F., Ronzano, F., & Saggion, H. (2014). Italian irony detection in Twitter: A first approach. *The First Italian Conference on Computational Linguistics CLiC-it 2014* (pp. 28–32). Italy: PISA.

Bosco, C., Lai, M., Patti, V., Virone, D. (2016). Tweeting and being ironic in the debate about a political reform: The French annotated corpus Twitter-mariagePourTous. In Calzolari, N., Choukri, K., Declerck, T., Uğur Doğan, M., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (Eds.), *Proceedings of the 10th international conference on language resources and evaluation (LREC 2016)*, ELRA, Portorož, Slovenia

Bouazizi, M., Ohtsuki, T. (2016). Sarcasm detection in Twitter: "all your products are incredibly amazing!!!"—are they really? In *Global Communications Conference, GLOBECOM 2015*. IEEE.

Brown, P., & Levinson, S. C. (1987). *Politeness: Some universals in language usage*. Cambridge: Cambridge University Press.

Burgers, C. (2010). Verbal irony: Use and effects in written discourse. PhD thesis, UB Nijmegen.

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., de Oliveira, E. (2009). Clues for detecting irony in user-generated contents: Oh...!! It's "So Easy" ;-). In *Proceedings of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion, TSA '09* (pp. 53–56). ACM, Hong Kong, China.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27:1–27:27.

Charalampakis, B., Spathis, D., Kouslis, E., & Kermanidis, K. (2016). A comparison between semi-supervised and supervised text mining techniques on detecting irony in greek political tweets. *Engineering Applications of Artificial Intelligence*, *51*(Supplement C), 50–57.

Chollet F, et al (2015) Keras. https://github.com/fchollet/keras.

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

Currie, G. (2006). Why irony is pretence. In S. Nichols (Ed.), *The architecture of the imagination: New essays on pretence, possibility, and fiction*. Oxford: Clarendon Press.

Davidov, D., Tsur, O., & Rappoport A. (2010). Semi-supervised recognition of sarcastic sentences in Twitter and amazon. In *Proceedings of the 14th conference on computational natural language learning, CoNLL'10* (pp. 107–116). ACL, Uppsala, Sweden.

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

Gers, F. A., & Schmidhuber, J. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. *Transactions on Neural Networks*, *12*(6), 1333–1340.

Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., & Reyes, A. (2015). SemEval-2015 task 11: Sentiment analysis of figurative language in Twitter. In *Proceedings of the 9th international workshop on semantic evaluation, SemEval'15* (pp. 470–478). ACL, Denver, Colorado.

Ghosh, A., & Veale, T. (2016). Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis* (pp. 161–169). ACL, San Diego, California.

Gibbs, R. W., O'Brien, J. E., & 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., et al. (2011). Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, HLT'11* (pp. 42–47). ACL, Portland, Oregon.

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

González-Ibáñez, R., Muresan, S., & Wacholder, N. (2011). Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, HLT'11* (pp. 581–586). ACL, Portland, Oregon.

Grice, P. H. (1978). Further notes on logic and conversation. In P. Cole (Ed.), *Syntax and semantics* (Vol. 9, pp. 113–127). New York: Academic Press.

Grice, P. H. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics. Speech acts* (Vol. 3, pp. 41–58). New York: Academic Press.

Heafield, K., Pouzyrevsky, I., Clark, J. H., & Koehn, P. (2013). Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st annual meeting of the association for computational linguistics* (pp. 690–696). ACL, Sofia, Bulgaria.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hogenboom, A., Bal, D., Frasincar, F., Bal, M., De Jong, F., & Kaymak, U. (2015). Exploiting emoticons in polarity classification of text. *Journal of Web Engineering*, *14*(1–2), 22–40.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD04* (pp. 168–177). ACM, Seattle, WA, USA.

Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys*, *50*(5), 73:1–73:22.

Joshi, A., Sharma, V., & Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on NLP* (pp. 757–762). ACL, Beijing, China.

Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P., & Carman, M. J. (2016). Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 conference on empirical methods in NLP* (pp. 1006–1011). ACL, Texas, USA.

Karoui, J., Benamara, F., Moriceau, V., Aussenac-Gilles, N., Hadrich Belguith, L. (2015). Towards a contextual pragmatic model to detect irony in tweets. In *Proceedings of the 53rd annual meeting of ACL and the 7th international joint conference on natural language processing of the Asian Federation of NLP* (pp. 644–650). ACL, Beijing, China.

Karoui, J., Benamara, F., Moriceau, V., Patti, V., Bosco, C., & Aussenac-Gilles, N. (2017). Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. In *Proceedings of the 15th EACL conference* (pp. 262–272). ACL, Valencia, Spain.

Kingma, D.P., & Ba, J.L. (2014). Adam: A method for stochastic optimization. CoRR abs/1412.6980

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

Kreuz, R. J., & 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., & 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., & van den Bosch, A. (2015). Signaling sarcasm: From hyperbole to hashtag. *Information Processing and Management*, *51*(4), 500–509.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159.

Liu, B. (2012). *Sentiment analysis and opinion mining*. San Rafael: Morgan & Claypool Publishers.

Lunando, E., & Purwarianti, A. (2015). Indonesian social media sentiment analysis With sarcasm detection. CoRR abs/1505.03085

Maynard, D., & Greenwood, M. (2014). Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the 9th international conference on language resources and evaluation* (pp. 4238–4243). ELRA, Reykjavik, Iceland.

McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. CoRR abs/1708.00107

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*. Red Hook: Curran Associates Inc.

Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the second joint conference on lexical and computational semantics (*SEM), volume 2: Proceedings of the 7th international workshop on semantic evaluation, SemEval'13* (pp. 321–327). ACL, Atlanta, Georgia, USA.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov. V. (2016). SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)* (pp. 1–18). ACL, San Diego, California.

Nielsen, F. A. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In Rowe M, Stankovic M, Dadzie AS, Hardey M (Eds.) *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages* (vol. 718, pp. 93–98). CEUR-WS.org, Heraklion, Crete, CEUR.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (vol. 14, pp. 1532–1543). ACL, Doha, Qatar.

Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A Deeper look into sarcastic tweets using deep convolutional neural networks. CoRR abs/1610.08815.

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

Reyes, A., Rosso, P., & 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., & Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the conference on empirical methods in natural language processing, EMNLP'13* (pp. 704–714). ACL, Seattle, Washington, USA.

Ritter, A., Clark, S., & Mausam, Etzioni O. (2011). Named entity recognition in tweets: An experimental study. In *Proceedings of the conference on empirical methods in natural language processing, EMNLP'11* (pp. 1524–1534). ACL, Edinburgh, United Kingdom.

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

Sperber, D., & Wilson, D. (1981). Irony and the use: Mention distinction. In P. Cole (Ed.), *Radical pragmatics* (pp. 295–318). New York: Academic Press.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J. (2012). BRAT: A web-based tool for NLP-assisted text annotation. In *Proceedings of the 13th conference of the European chapter of the association for computational linguistics, EACL'12* (pp. 102–107). ACL, Avignon, France.

Stone, P. J., Dunphy, D. C. D., Smith, M. S., & Ogilvie, D. M. (1966). *The general inquirer: A computer approach to content analysis*. Cambridge: The MIT Press.

Stranisci, M., Bosco, C., Irazú Hernández Farías, D., Patti, V. (2016). Annotating sentiment and irony in the online Italian political debate on #labuonascuola. In Calzolari N, Choukri K, Declerck T, Loftsson H, Maegaard B, Mariani J, Moreno A, Odijk J, Piperidis S (Eds.), *Proceedings of the 10th international conference on language resources and evaluation (LREC'16)*. ELRA, Portorož, Slovenia.

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

Van Hee, C., Lefever, E., & Hoste, V. (2016a). Exploring the realization of irony in Twitter data. In *Proceedings of the 10th international conference on language resources and evaluation, LREC'16* (pp. 1795–1799). ELRA, Portorož, Slovenia.

Van Hee, C., Lefever, E., & Hoste, V. (2016b). Guidelines for annotating irony in social media text, version 2.0. Tech. Rep. 16-01, LT3, Language and Translation Technology Team–Ghent University.

Veale, T., & Hao, Y. (2009). Support structures for linguistic creativity: A computational analysis of creative irony in similes. In *Proceedings of CogSci* (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., & Sperber, D. (1992). On verbal irony. *Lingua*, *87*(1), 53–76.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing, HLT'05* (pp. 347–354). ACL, Vancouver, Canada.