We Usually Don't Like Going to the Dentist: Using Common Sense to Detect Irony on Twitter

Cynthia Van Hee
Ghent University
LT3, Faculty of Arts and Philosophy
cynthia.vanhee@ugent.be

Els Lefever
Ghent University
LT3, Faculty of Arts and Philosophy
els.lefever@ugent.be

Véronique Hoste
Ghent University
LT3, Faculty of Arts and Philosophy
veronique.hoste@ugent.be

Although common sense and connotative knowledge come naturally to most people, computers still struggle to perform well on tasks for which such extratextual information is required. Automatic approaches to sentiment analysis and irony detection have revealed that the lack of such world knowledge undermines classification performance. In this article, we therefore address the challenge of modeling implicit or prototypical sentiment in the framework of automatic irony detection. Starting from manually annotated connoted situation phrases (e.g., "flight delays," "sitting the whole day at the doctor's office"), we defined the implicit sentiment held towards such situations automatically by using both a lexico-semantic knowledge base and a data-driven method. We further investigate how such implicit sentiment information affects irony detection by assessing a state-of-the-art irony classifier before and after it is informed with implicit sentiment information.

1. Introduction

With the advent of the Web 2.0, information 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 and leave obvious things unstated. As such, the utterance "He lacks social responsibility"

Submission received: 13 October 2017; revised version received: 9 May 2018; accepted for publication: 20 August 2018.

doi:10.1162/coli_a_00337

will be perceived as negative, because it is obvious for most people that social responsibility is a positive human quality. These obvious things are part of **common sense**: knowledge that people have of the world they live in, and that serves as a basis to form judgments and ideas (Cambria et al. 2009).

Although this commonsense knowledge often refers to the obvious factual things people normally know about the world they live in, it can also refer to the affective information associated with these real-world events, actions, or objects. This connotative knowledge, or typical sentiment related to real-world concepts, comes naturally to most people, but is far from trivial for computers. Perhaps the most salient example of this are sentiment analysis systems, which show good performance on explicit sentiment expressions (e.g., "brilliant" in Example (1) (e.g., Van Hee et al. 2014; Deriu et al. 2016; Mohammad, Sobhani, and Kiritchenko 2016), but struggle with text fragments that involve implicit sentiment (e.g., "shuts down at 40%" in Example (2).

(1) 3000th tweet dedicated to Andy Carroll and West Ham, brilliant start to the season!

(2) Since last update iPhone 6 battery shuts down at 40%.

Such implicit sentiment or connotative knowledge (i.e., the feeling a concept generally invokes for a person or a group of people) is also referred to as **prototypical sentiment** (Hoste et al. 2016). Like in Example (2), prototypical sentiment expressions are devoid of subjective words and rely on common sense shared by the speaker and receiver in an interaction. To be able to grasp such implied sentiment, sentiment analysis systems require additional knowledge that provides insight into the world we live in and affective information associated with natural language concepts.

Although modeling implicit sentiment is still in its infancy (Cambria et al. 2016), such linking of concepts or situations to implicit sentiment will open new perspectives in natural language processing (NLP) applications, not only for sentiment analysis tasks, but also for any type of tasks that involves semantic text processing, such as automatic irony detection and the detection of cyberharassment (Dinakar et al. 2012; Van Hee et al. 2015). Although automatic systems perform well in deriving particular information from a given text (i.e., the meaning of a word in context, emotions expressed by the author of a text), they often struggle to perform tasks where extratextual information is necessary to grasp the meaning of an utterance.

As mentioned earlier, one typical application where the lack of such implicit information becomes apparent is automatic irony detection. **Irony** is traditionally defined as a rhetorical device where an evaluative utterance expresses the opposite of what is actually intended (e.g., Grice 1978; Burgers 2010; Camp 2012). This implies the expression of a positive evaluation when a negative one is intended, or vice versa. By doing so, speakers often rely on un-uttered knowledge such as mutually shared information or world knowledge. An example of such implied (subjective) information is contained in the ironic utterance "Cannot wait to go to the dentist tomorrow!". The apparent positive polarity expressed by "cannot wait" is contrasted with the negative sentiment implied by a visit to the dentist. The latter can be considered a prototypical unpleasant activity; that is, even without explicit sentiment words, users understand its negative connotation. Observing this contrast enables its listener to infer that the utterance is meant ironically. The human brain excels at understanding such implicit meanings, as people learn from experiences and feel certain emotions based on appraisals (Scherer 1999) and inferences from related experiences. Computers, by contrast, lack such world knowledge and can only rely on what they have learned from specific data.

In fact, when modeling subjectivity and sentiments in text, commonly used approaches include lexicon-based and statistical or machine learning methods (Liu 2015; Cambria et al. 2017). The majority of these approaches focus on identifying explicit sentiment clues in text: Lexicon-based approaches make use of sentiment dictionaries (e.g., Wilson, Wiebe, and Hoffmann 2005; Mohammad and Turney 2013) and machine learning sentiment classifiers generally exploit features that represent explicit sentiment clues, like bags-of-words, punctuation marks, flooded characters, and so forth. Much more challenging is detecting implicit sentiment, or identifying non-evaluative words that evoke a particular sentiment.

In this article, we confront the challenge of automatically recognizing implicit sentiment in tweets and we explore whether such implicit sentiment information benefits automatic irony detection. Several studies have underlined the importance of implicit sentiment for irony detection (e.g., Riloff et al. 2013; Wallace 2015); here, we present, to our knowledge, the first attempt to model implicit sentiment by using both a lexico-semantic knowledge base and a data-driven approach based on real-time tweets. Moreover, manual annotations of our irony data set allowed us to evaluate the approach using gold-standard connoted situations such as *going to the dentist*. Next, the validity of our approach to model implicit sentiment is assessed by evaluating the performance of a state-of-the-art irony detection system before and after informing it with implicit sentiment information.

The remainder of this article is structured as follows: Section 2 presents an overview of related research on irony detection and implicit sentiment modeling, and Sections 3, 4, and 5 zoom in on the different experimental set-ups. In Section 3, we present a state-of-the-art irony detection system and in Section 4 we investigate the feasibility to model implicit sentiment in an automatic way. Exploring whether implicit sentiment information benefits irony detection is the focus of Section 5. Section 6 concludes and suggests some directions for future research.

2. Modeling Implicit Sentiment

Modeling implicit sentiment is not a new challenge. Efforts to tackle this problem have been undertaken in different research areas, among others, sentiment analysis, content analysis in journalism, and irony detection. Although early work by Lin et al. (2006) investigated how to identify the perspective from which a document is written automatically, it is Greene (2007) and Wilson (2008) who have pioneered implicit sentiment research. Greene introduced the concept of what he later called **syntactic packaging** (Greene and Resnik 2009) and demonstrated the influence of syntactic choices on the perceived implicit sentiment of news headlines. He showed, for instance, that the active voice tends to attribute a greater sense of responsibility to the agent of a sentence, causing "a soldier veered his jeep into a crowded market and killed three civilians" to be perceived as more negative toward the soldier than "a soldier's jeep veered into a crowded market, causing three civilian deaths." Wilson annotated **objective polar utterances** or statements that describe positive or negative factual information about something (e.g., "The camera broke the first time I used it") in meeting report content. A similar study was conducted by Toprak, Jakob, and Gurevych (2010), who annotated objective polar utterances in consumer reviews, but named them **polar facts**. Deng and Wiebe (2014) detected implicit sentiment by inferencing over explicit sentiment expressions, namely, through implicature rules (i.e., "goodFor" and "badFor"), describing events that have either a positive or negative effect on objects or entities. Ebrahimi

(2013) tackled this problem in the medical domain and exploited disease symptoms as negative implicit sentiment features to predict side effects in drug reviews.

Implicit or prototypical sentiment is part of common sense, meaning that the information is not explicitly mentioned, but implicitly shared by the participants in a conversation. It is important to note, however, that this implied sentiment does not apply for every individual: Although studying chemistry all weekend may have an implied negative sentiment for most people, it is highly probable that some individuals would love to study chemistry all weekend.

To make such information available to machines, the past few years have witnessed a number of efforts to construct common sense databases. Among the most well-known are probably Cyc (Lenat 1995), WordNet (Miller 1995), FrameNet (Fillmore, Johnson, and Petruck 2003), and DBPedia (Lehmann et al. 2015). These 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, such as $\langle dentist \rangle$ *is_a* $\langle doctor \rangle$. For sentiment analysis, however, there is an additional need for knowledge about the typical sentiments people hold toward 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. In SentiWordNet, each WordNet synset is associated with three numerical scores describing how objective, positive, and negative the terms in the synset are. Finally, SenticNet (Cambria et al. 2016) is a knowledge and *sentics* database aiming to make conceptual and affective information more easily accessible to machines. Mainly built upon ConceptNet, the knowledge base contains commonsense information for 50,000 concepts and outperforms other resources for sentiment analysis tasks (Cambria et al. 2016).

Seminal work on constructing similar knowledge bases has also been done on a smaller scale. For instance, Balahur et al. (2011) constructed a knowledge base containing situation descriptions that evoke particular emotions. They built EmotiNet, a database containing <actor – action – patient – emotion> tuples from the ISEAR-corpus,[2] a collection of self-reported situations in which respondents experienced particular emotions such as joy, anger, and disgust. The database was expanded by extracting connotative information from Twitter (i.e., bootstrapping by using seed words like "failure" and "disease"), and using ConceptNet (Speer and Havasi 2013). The development of EmotiNet is grounded in Appraisal Theory (Scherer 1999) and aims to store emotional reactions to real-world contexts (e.g., "I'm going to a family party because my mother obliges me to" $\rightarrow$ disgust). The knowledge base was used to learn intrinsic properties of entities and situations in journalistic text that trigger certain emotions like "refugees" and "being sentenced" (Balahur and Tanev 2016). Feng et al. (2013) created a connotation lexicon that determines the underlying sentiment of seemingly objective words (e.g., cooperation$^{(+)}$, overcharge$^{(-)}$) and the general connotation of named entities (e.g., Einstein$^{(+)}$, Osama$^{(-)}$). Zhang and Liu (2011) hypothesized that resource phrases (e.g., "this washer uses a lot of electricity") are important carriers of implicit sentiment. They automatically extracted resource terms like "water" and "money" with resource

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

2 International Survey on Emotion Antecedents and Reactions corpus.

usage verbs such as "use" and "spend" and found that, when occurring together, they often imply a positive or negative sentiment. To be able to identify implicit sentiment expressions below the sentence level, Van de Kauter, Desmet, and Hoste (2015) developed the first fine-grained annotation scheme for implicit sentiment in financial newswire text. The scheme is able to pinpoint phrases that express explicit (so-called **private states**) and implicit (**polar facts**) sentiment. They demonstrated the validity of the fine-grained sentiment annotations by comparing them to two baseline sentiment classification methods (i.e., lexicon- and machine-learning-based) for sentiment analysis in financial news text and found that the former largely outperformed the baselines (Van de Kauter, Breesch, and Hoste 2015).

These studies have mostly tackled implicit sentiment modeling to improve automatic sentiment analysis. Research on automatic irony detection, however, has also uncovered the need to model common sense. Among other researchers, Riloff et al. (2013) and Van Hee, Lefever, and Hoste (2016b) demonstrated that a common form of irony includes an utterance in which a positive evaluation (e.g., "cannot wait" or "so happy") targets a prototypical negative situation (e.g., "go to the dentist," "flight delay"). Whereas the former can mostly be identified using sentiment lexicons, the latter requires implicit sentiment resources, which are (i) much more scarce and (ii) often tailored toward a specific domain—for instance, family situations (Balahur et al. 2011). To tackle this problem, Riloff et al. bootstrapped negative situation phrases in the vicinity of positive seed words like "love." They showed that such seed words are useful to find prototypical negative concepts (e.g., "working" was learned as a negative situation as it followed "I love" in ironic text) and showed that integrating polarity contrasts based on implicit sentiment outperformed the *n*-gram baseline for irony detection with three points. The authors pointed, however, to some important restrictions of their approach, namely, that (i) only negative verb phrases were considered as connoted situations, and (ii) attached (prepositional) phrases were not captured. For instance, in the phrases "working on my last day of summer" and "working late 2 days in a row," only "working" was considered a negative situation. Comparable research was done by Joshi, Sharma, and Bhattacharyya (2015), who used lexical and pragmatic features for irony detection, but also integrated explicit and implicit incongruity features. They applied the same strategy as Riloff et al. to infer implicit sentiment, but included positive situation phrases as well and retained subsumed polarity phrases (e.g., "working late 2 days in a row"). Their system outperformed a baseline exploiting merely lexical features with approximately four points on a corpus of hashtag-labeled tweets (i.e., tweets with a *#sarcasm* hashtag were considered ironic, tweets devoid of such a hashtag were not ironic), but it did not outperform the baseline on a smaller, manually labeled irony corpus.

Both of these studies take a bootstrapping approach to model implicit sentiment. They make use of polar patterns (e.g., "I love [...]", "[...] makes me sick") to extract connoted situation phrases with opposite polarity. When extracting *n*-grams nearby a positive polar expression, they take its negative implicit polarity for granted. In this article, we manually annotated such situation phrases and propose a method to define the polarity of the phrases automatically in an open setting (i.e., without domain restrictions or seed words) by analyzing how people talk about the situations on Twitter. The above-mentioned studies use Twitter as the source of implicit or prototypical sentiment, but the bootstrapping algorithm collects information from a single tweet. For instance, if the algorithm encounters the tweet "How I love sunny mornings #sarcasm," "sunny mornings" is added to the negative situation phrases as it follows the ironic utterance "how I love." In this study, on the contrary, we infer prototypical sentiment

toward a situation as derived from a collection of tweets, and, therefore, felt by multiple people.

Karoui et al. (2015) hypothesized that factual oppositions (i.e., 'not P' while it can be verified that P is true) are good indicators for irony. Their automatic irony detection approach consisted in three steps: First, they identified, based on a rich feature set, whether a tweet is ironic. Second, if a tweet is classified as non-ironic, they searched for negated facts, such as "#Valls is not the interior minister" in the tweet "#Valls has learnt that Sarkozy was wiretapped in newspapers. Fortunately he is not the interior minister." Those facts were verified using external resources like Wikipedia. Third, if the facts were found to be true (i.e., Valls is the interior minister), then the tweet's initial classification label was changed to "ironic." This way, the authors proposed a novel method to integrate pragmatic context from external resources to detect the irony. Our approach is similar in that we also make use of extra-textual information, to infer implicit sentiment or common sense. However, we believe we take a more inclusive approach to irony detection by focusing on polarity oppositions, which show to be very frequent in ironic tweets (see subsequent discussion). In fact, our corpus analysis reveals that factual oppositions only occur in one particular type of ironic tweets (i.e., *other verbal irony*) and represent a small portion (16%) of these tweets.

3. Automatic Irony Detection

In this section, we describe our baseline/preliminary methodology for automatic irony detection in English tweets. The main question we aim to answer here is, *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, including among others lexical, syntactic, and semantic features. The analysis of the experimental results reveals the need to also model implicit sentiment information, which will be investigated in the next section.

We start this section with a brief overview of the state of the art in irony detection.[3] Next, a detailed description of our irony modeling approach is given. This includes the presentation of a manually annotated irony corpus and details on how we developed our irony detection pipeline and experimented with different information sources for the task.

3.1 State of the Art in Irony Detection

Research in NLP has recently seen various attempts to tackle irony detection. As described by Joshi, Bhattacharyya, and Carman (2017), irony modeling approaches can roughly be classified into rule-based and machine learning methods. Whereas rule-based approaches mostly rely on lexical information and require no training (e.g., Veale and Hao 2010; Maynard and Greenwood 2014; Khattri et al. 2015), machine learning does utilize training data and exploits various information sources (or *features*), including bags of words, syntactic patterns, sentiment information, and semantic relatedness (Davidov, Tsur, and Rappoport 2010; Liebrecht, Kunneman, and van den Bosch 2013; Reyes, Rosso, and Veale 2013). Twitter has been a popular data genre for the task, as

3 When discussing related research, we refer to irony using the terminology utilized by the corresponding researchers (i.e., "sarcasm," "irony," or "verbal irony").

its self-describing hashtags (e.g., #*irony*) facilitate data collection. Many supervised learning approaches utilize irony-related hashtags as class labels, but as such labels have been shown to increase data noise (Kunneman et al. 2015; Van Hee 2017), manually labeled corpora have become increasingly important for irony detection (Riloff et al. 2013; Khattri et al. 2015; Van Hee 2017).

Machine learning approaches to detect irony vary in terms of features and learning algorithms. Early work by Davidov, Tsur, and Rappoport (2010) describes a semi-supervised approach exploiting punctuation and syntactic patterns as features. Similarly, Bouazizi and Ohtsuki (2016) extracted more than 300,000 part-of-speech patterns and combined them with lexical (e.g., bags of words), sentiment (e.g., number of positive words), and syntactic (e.g., number of interjections) features to train a random forest classifier. González-Ibáñez, Muresan, and Wacholder (2011) combined lexical with pragmatic features such as frowning emoji and @-replies. As classifiers they used sequential minimal optimization and logistic regression. Reyes, Rosso, and Veale (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. Kunneman et al. (2015) pioneered irony detection in Dutch tweets using word *n*-gram features and a Balanced Winnow classifier. Van Hee, Lefever, and Hoste (2016b) combined lexical with sentiment, syntactic, and semantic Word2Vec cluster features for irony detection using a support vector machine (SVM). Recent work by Ghosh and Veale (2016), Poria et al. (2016), and Zhang, Zhang, and Fu (2016) has approached irony detection using (deep) neural networks, which make use of continuous automatic features instead of manually defined ones.

Besides text-based features, irony research has explored extratextual features related to the author or context of a tweet, such as previous tweets or topics, author profile information, typical sentiment expressed by an author, and so on (Bamman and Smith 2015; Wang et al. 2015). Riloff et al. (2013), Khattri et al. (2015), and Van Hee (2017) exploited implicit or prototypical sentiment information to model a polarity contrast that characterizes ironic language.

3.2 Corpus Description

To operationalize irony detection, we constructed a data set of 3,000 English tweets by searching Twitter with the hashtags #*irony*, #*sarcasm*, and #*not*. For this purpose, we made use of Tweepy,[4] a Python library to access the official Twitter API. The tweets were collected between 1 December 2014 and 4 January 2015, represent 2,676 unique Twitter users, and have an average length of 15 tokens. An example tweet is presented in Figure 1. To minimize data noise (see earlier), all tweets were manually labeled using a newly developed fine-grained annotation scheme.

Although 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 (French, English, and Italian) ironic tweets and they defined eight fine-grained categories of pragmatic devices that realize such an incongruity, including analogy, hyperbole, rhetorical question, oxymoron, and so forth. The typology provides valuable insights into the linguistic realization of irony that could improve

4 https://github.com/tweepy/tweepy.

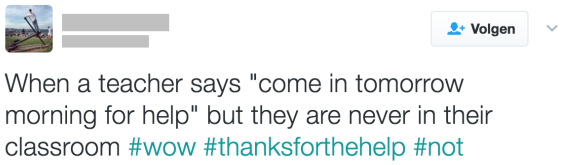


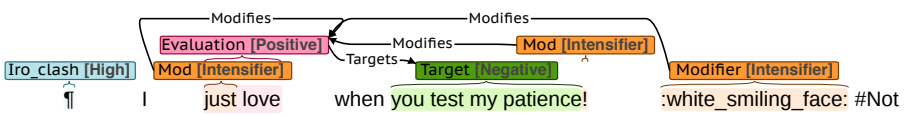
Figure 1
Corpus example.

its automatic detection (e.g., the correlation between irony markers and irony activation types). However, given the complexity of identifying such pragmatic devices as demonstrated by the inter-annotator agreement study, it is not clear to which extent it would be computationally feasible to detect the irony categories they propose. The annotation scheme that was applied to our corpus was also applied to multilingual data (Dutch, English) and distinguishes three forms of irony. We refer to Van Hee, Lefever, and Hoste (2016a) for a detailed overview of the annotation scheme, but present the main annotation categories here.

Literature states that irony is often realized by means of a polarity contrast (e.g., Grice 1975; Burgers 2010). As the starting point of the annotation process, we therefore define such 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, Lefever, and Hoste 2016a). To allow for other types of irony to be annotated as well, the annotation scheme describes three main categories:

1. **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.
2. **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.
3. **Not ironic:** The text is not ironic.

To better grasp the linguistic realization of ironic utterances, in the case of a tweet belonging to category 1, the annotators marked the relevant text spans and indicated whether the evaluations were positive or negative and whether they were explicit or implicit. Example (3) shows a tweet where a positive utterance ("just love") is contrasted with a negatively connoted situation ("you test my patience").

(3)


Such text spans that consist of factual information while evoking a specific sentiment are called **targets** throughout this article, as they present the target of the literal sentiment expression that is meant ironically. A post-annotation quantitative analysis of the implicit and explicit evaluations contained in this category revealed that, in a total of 1,728 *ironic by clash* utterances, 1,909 explicit positive (e.g., "AMAZING!") evaluations

Table 1
Inter-annotator agreement (Fleiss' Kappa) obtained in two annotation rounds.

annotation	Kappa κ round 1	Kappa κ round 2
ironic / not ironic	0.65	0.72
ironic by clash / other / not ironic	0.55	0.72
hashtag indication	0.60	0.69
harshness	0.32	0.31
polarity contrast (target – evaluation)	0.66	0.55

were annotated, and 501 explicit negative ones (e.g., "#losers"). When looking at implicit evaluations in the tweets, we found 13 annotated targets with a positive implicit sentiment (e.g., "the wake-up wrap and coffee this morning") and 836 targets that were assigned a negative implicit polarity (e.g., "homework in the weekends").

The corpus was entirely annotated by three Masters students in linguistics and second-language speakers of English, with each student annotating one third of the whole corpus. To assess the reliability of the annotations, an *inter-annotator agreement study* was set up in two rounds. First, inter-rater agreement was calculated between the authors of the guidelines. The aim of this 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. After adding some refinements to the scheme (Van Hee (2017), a second agreement study was carried out by three Master's students, who each annotated the same subset (i.e., 100 randomly selected instances) of the corpus.

In both rounds, inter-annotator agreement was calculated at different steps in the annotation process. As the 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). The results of the inter-annotator agreement study are presented in Table 1. With the exception of harshness, which proves to be difficult to judge on, Kappa scores show a moderate to substantial agreement between annotators for the different annotation steps.[5] Overall, we see that similar or better agreement was obtained after the refinement of the annotation scheme, which had the largest effect on the irony annotation. 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 round. An explanation for this drop is that one out of the three annotators in the second round performed less well than the others for this particular annotation. In fact, annotator disagreement mostly concerned the annotation of implicit sentiment (i.e., targets). This makes intuitive sense, as the annotation is (even unconsciously) more subject to personal preferences, as opposed to explicit sentiment expressions. The following is an example for which disagreement between the annotators was observed:

(4) Hey there! Nice to see you Minnesota/ND Winter Weather #Not

In this above tweet, two out of the three annotators indicated "Winter Weather" as a negative target (i.e., carrying an implicit negative sentiment) that contrasts with the

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

positive expression "Nice to see you." The third annotator, however, indicated that the hashtag "#Not" was necessary to understand the irony in this tweet. After discussion with the annotators and experts, all annotators agreed on the implicit negative polarity associated with winter weather.

Given the difficulty of the task, a Kappa score of 0.72 for recognizing 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. Consequently, once the annotation of the entire corpus was completed, all annotations of implicit sentiment were reconsidered by one of the experts.

Out of the total of 3,000 tweets we collected with *#irony*, *#sarcasm*, and *#not*, 604 were considered not ironic. This would mean that a hashtag-labeled irony corpus can contain about 20% noise. This is twice the amount of noise that was observed in a similar study by Kunneman et al. (2015). There are several possible explanations for this noise, such as the grammatical use of *#not* as a negator instead of an irony indicator, the metalinguistic use of such hashtags (e.g., "I love that his humor is filled with #irony"), and misuse of the hashtags by adding it to non-ironic content. Such non-ironic tweets were added to the negative class for our irony detection experiments, which leaves 2,396 ironic and 604 non-ironic tweets in the experimental corpus. To balance the class distribution in this corpus, we expanded the latter with a set of 1,792 non-ironic tweets from a background Twitter corpus. The tweets in this data set were collected from the same set of Twitter users as in the irony corpus (henceforth referred to as the *hashtag corpus*), and within the same time span. After collection, all tweets containing irony-related hashtags were automatically removed and an additional manual filtering was done to make sure that no ironic tweets were contained in the corpus.

Table 2 presents the experimental corpus comprising different irony categories as annotated in the hashtag corpus and 1,792 non-ironic tweets from a background corpus that were included to obtain a balanced class distribution, resulting in a total corpus size of 4,792 tweets. For the experiments, the corpus was randomly split into a balanced training and test set of, respectively, 80% (3,834 tweets) and 20% (958 tweets). The former was used for feature engineering and classifier optimization purposes, and the latter functioned as a held-out test set to evaluate and report classification performance.

3.3 Preprocessing and Feature Engineering

Prior to feature extraction and training of the model, the experimental corpus was preprocessed. As preprocessing steps, we applied tokenization, part-of-speech tagging, lemmatization, and named entity recognition. Tokenization and part of speech-tagging were done using the Carnegie Mellon University Twitter NLP Tool (Gimpel et al. 2011),

Table 2
Experimental corpus statistics: Number of instances per annotation category plus non-ironic tweets from a background corpus.

	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**	

which is trained on user-generated content. For lack of a reliable Twitter-specific lemmatizer, we made use of the LeTs Preprocess toolkit (Van de Kauter et al. 2013). We used the Twitter named entity recognizer by Ritter et al. (2011) for named entity recognition.

Additionally, all tweets were cleaned (e.g., replacement of HTML-escaped characters) and a number of (shallow) normalization steps were introduced to decrease feature sparseness. In concrete terms, all hyperlinks and @-replies in the tweets were normalized to "http://someurl" and "@someuser," respectively, and abbreviations were replaced by their full form, based on an English abbreviation dictionary[6] (e.g., "w/e" → "whatever"). Furthermore, variations in suspension dots were normalized to three 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, we extracted a number of features to train our irony detection system. Based on the information they provide, the features can be divided into four groups, namely, **lexical**, **syntactic**, **sentiment**, and **semantic** features. The feature groups bring together a varied set of information sources, most of which have proven their relevance for this type of task in other studies as well, including bags of words (e.g., Liebrecht, Kunneman, and van den Bosch 2013), part-of-speech information (e.g., Reyes and Rosso 2012), punctuation and word-shape features (e.g., Tsur, Davodiv, and Rappoport 2010), interjections and polarity imbalance (e.g., Buschmeier, Cimiano, and Klinger 2014), sentiment lexicon features (e.g., Bouazizi and Ohtsuki 2016), and semantic similarity based on Word2Vec embeddings (Joshi et al. 2016). The following paragraphs present a detailed overview of the different feature groups that were defined.

3.3.1 Lexical Features. A first set of lexical features are **bags of words** (*bow*) or *n*-grams formed by words and characters extracted from the training corpus. Based on preliminary experiments on our data set, word unigrams and bigrams (*w1g, w2g*), as well as character trigrams and fourgrams (*ch3g, ch4g*), were extracted as binary, sparse features. The *n*-grams were created using raw words rather than lemmas or canonical forms to retain morphological information,[7] 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.

Second, the lexical features contain a set of **word form features**: character and punctuation flooding, punctuation last token, number of punctuation marks/capitalized words/hashtag words/interjections, hashtag-to-word ratio, emoticon frequency, and tweet length. The first three are binary features, and the others are numeric and present normalized floats (i.e., divided by the tweet length in tokens), except the *tweet length* feature.

A third set of lexical features include **conditional *n*-gram probabilities** based on language model probabilities. Although often exploited in machine translation research (e.g., Bojar et al. 2016), language model information incorporated as features is, to our knowledge, novel in irony detection. The models were created with KENLM (Heafield et al. 2013) and are trained on an ironic and a non-ironic background corpus.[8] As

6 `http://www.chatslang.com/terms/abbreviations`.

7 *n*-grams based on raw tokens were preferred over lemma forms, as preliminary experiments revealed that better results were obtained with the former.

8 The data (1,126,128 tweets) were collected between April 2016 and January 2017 by crawling Twitter at regular intervals using the Twitter Search API. There is no overlap with the training corpus.

features we extracted log probabilities indicating how probable a tweet is likely to appear in either an ironic or non-ironic corpus. Two additional features include the number of out-of-vocabulary words (i.e., words that appear in the tweet, but that were not seen in the training corpus) a tweet contains based on each language model.

3.4 Syntactic Features

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

3.5 Sentiment Lexicon Features

Six sentiment lexicon features were extracted, indicating the number of positive/negative/neutral lexicon words in a tweet, the overall tweet polarity, the difference between the highest positive and lowest negative sentiment values found in a tweet, and a binary feature indicating the presence of a polarity contrast between two lexicon words. We made use of existing sentiment lexicons for English: AFINN (Nielsen 2011), General Inquirer (Stone et al. 1966), the MPQA subjectivity lexicon (Wilson, Wiebe, and Hoffmann 2005), the NRC emotion lexicon (Mohammad and Turney 2013), and Bing Liu's opinion lexicon (Liu, Hu, and Cheng 2005). All of the aforementioned lexicons are commonly used in sentiment analysis research (Cambria et al. 2017) and their validity has been confirmed in earlier experiments (Van Hee et al. 2014) where a preliminary study revealed that, by using merely these 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.

Sentiment lexicon features were extracted in two ways: (i) by considering all tweet tokens and (ii) by taking only hashtag tokens into account, after removing the hashtag (e.g., "lovely" from "#lovely"). Negation clues were taken into account by flipping the polarity of a sentiment word when it was preceded by a negation word (e.g., "not," "never," "don't").

3.6 Semantic Features

Our hypothesis is that ironic tweets may 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 utilized 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 a tweet. An example cluster is presented in Example (5).

Table 3
Feature statistics per feature group.

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

(5) `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 1,126,128 (ironic + non-ironic) tweets.[9] 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.

3.7 Feature Statistics

In summary, four feature groups were defined for the experiments, the statistics of which are presented in Table 3. As is considered good practice when working with SVM, the feature vectors were scaled prior to constructing models, meaning that all features were linearly mapped to the range [0,1]. As stated by Hsu, Chang, and Lin (2003), important advantages of feature scaling 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.

3.8 Experimental Design and Results

For the experiments, we made use of an SVM, as implemented in the LIBSVM library (Chang and Lin 2011). We chose an SVM as the classification algorithm because it has been successfully implemented with large feature sets and because its performance for similar tasks has been recognized (e.g., Riloff et al. 2013; Joshi, Bhattacharyya, and Carman 2017).

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 data set showed even better results using RBF. Given the importance of parameter optimization 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 parametrization, γ was varied between 2^{-15} and 2^{3} (stepping by factor 4), and 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 set-up using all the training data, which was evaluated on the held-out test set.

9 The same corpus is used as for the n-gram probabilities in the lexical feature set (see above).

As the evaluation metrics, we report accuracy (the number of correct predictions divided by the total number of predictions), precision (the proportion of the data points the model says was relevant actually were relevant), recall (the proportion of relevant instances that were retrieved), and F_1 score (the harmonic mean of precision and recall), indicating how well the classifier detects irony. While the latter represents an average of the F_1 scores per class when used in multi-label or multi-class classification, in binary classification or detection tasks like the present, it is calculated on the positive (i.e., ironic) instances only.

It is important to note that the experimental results we present in the following sections are calculated on the held-out test set. In-between results obtained through cross-validation on the development set are not included because of space constraints.

3.8.1 Baselines. 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 randomly assigns a class label (i.e., ironic or not ironic) to each instance. Next, we calculated the performance of two classifiers, one based on token unigram (*w1g*) and bigram (*w2g*) features, and a second one using character trigram (*ch3g*) and fourgrams (*ch4g*) features. Hyperparameter optimization is crucial to the good functioning of the algorithm; hence it was also applied in the baseline experiments, except for *random class*.

Table 4 displays the baseline scores on the held-out test set. Although the random class baseline clearly benefits from the balanced class distribution, we find that the *n*-gram classifiers already present strong baselines for the task. In fact, earlier studies showed that *n*-gram features have proven to work well for this task, despite their simplicity and universal character (e.g., Liebrecht, Kunneman, and van den Bosch 2013; Reyes, Rosso, and Veale 2013).

3.8.2 Individual Feature Groups. Having established the baselines, we 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 5 displays the scores of the individual feature groups on the held-out test set. To facilitate comparison, the baseline scores are included in gray. The best results per column are indicated in bold.

Table 5 confirms the strong baseline that present *n*-gram features, given that none of the feature groups outperforms the character *n*-gram baseline in terms of F_1 score. In terms of recall, syntactic features score better. Character *n*-gram features outperforming the lexical feature group (which contains a fair number of other lexical clues in addition to character *n*-grams) seems to indicate that the former work better for irony detection. This seems counterintuitive, however, because the lexical feature group includes information that 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

Table 4
Classification results of the baselines (obtained on the test set).

baseline	accuracy	precision	recall	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 5
Irony detection results (obtained on the held-out test set) using 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%

individual features in the lexical feature group (potentially the most informative ones) is undermined by the feature abundance in the group. Lexical features are, however, the only ones that outperform the token *n*-gram baseline. Although 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, and that these features are not directly derived from the training data, as opposed to bag-of-words features. Recall being less than 50% for the sentiment lexicon features shows that, when using merely explicit sentiment clues, about half of the ironic tweets are missed by the classifier. This observation is in line with the findings of Riloff et al. (2013), who report irony detection scores between $F_1 = 14\%$ and $F_1 = 47\%$ when using merely sentiment lexicons.

Based on the results, we conclude that overall, lexical features perform best for the task ($F_1 = 67\%$). However, the best recall (69%) is obtained using syntactic features. 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 specific type or realization of irony. We observed, for instance, that lexical features are strongly predictive of irony (especially *ironic by clash*) in short tweets and tweets containing exaggerations (e.g., character repetition, see Example (6)), while sentiment features often capture *ironic by clash* instances that are very subjective or expressive (Example (7)). Syntactic features seem to work well for predicting irony in long tweets and tweets containing *other verbal irony* (Example (8)). Finally, semantic features contribute most to detecting situational irony (Example (9)).

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

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

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

(9) SO there used to be a crossfit place here ... #pizzawins

3.8.3 Feature Group Combinations. The previous paragraphs showed 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 of irony as well. This is why we investigate in this section the potential of combining the aforementioned feature groups. Table 6 presents the results of a binary irony classifier exploiting a combination

Table 6
Irony detection results (obtained on the held-out test set) using 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%

of feature groups obtained on the held-out test set. The best individual feature group (i.e., lexical) and the character *n*-gram baselines are also included for the purpose of comparison. From the results in Table 6, 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%.

3.8.4 Analysis. In Table 7, we compare the results of our best SVM-classifier (i.e., exploiting lexical + semantic + syntactic features) with that of state-of-the-art irony detection approaches. All scores are obtained on the test set created by Riloff et al. (2013), which originally consisted of 3,000 manually annotated tweets, 690 (or 23%) of which were *sarcastic*. However, the reported scores only apply to a subset of the corpus, because of the perishability of Twitter data (i.e., only tweet IDs could be provided to download the actual content of the tweets).

Table 7
Comparison of our approach to three state-of-the-art irony detection methods. The results are obtained on Riloff et al.'s (2013) irony data set. The best results per column are in **bold**.

approach	corpus size	precision	recall	F_1
Farías, Patti, and Rosso (2016)	474 ironic + 1,689 non-ironic	–	–	**73.00%**
Joshi, Sharma, and Bhattacharyya (2015)	506 ironic + 1,772 non-ironic	**77.00%**	51.00%	61.00%
Riloff et al. (2013)	693 ironic + 2,307 non-ironic	62.00%	44.00%	51.00%
Our approach	401 ironic + 1521 non-ironic	63.54%	**70.35%**	66.78%

Table 7 presents precision, recall, and F_1 scores (except for Farías, Patti, and Rosso [2016], who only report F_1 scores) obtained by three state-of-the-art irony detection systems on the same data set. Our approach outperforms that of Riloff et al. (2013) and Joshi, Sharma, and Bhattacharyya (2015) in terms of F_1 score, but not that of Farías, Patti, and Rosso (2016). However, it is important to note that the results in the table should be interpreted carefully. Our approach reports macro-averaged scores to assign each class equal weight in the evaluation (thus giving equal weight to the minority positive—i.e., ironic class) and Joshi, Sharma, and Bhattacharyya (2015) report that they applied weighted averaging. It is not clear, however, whether and which averaging method was applied by Riloff et al. (2013) and Farías, Patti, and Rosso (2016). As scores may vary according to how they are averaged (e.g., if micro-averaged, more weight is given to the negative class), such information is required to allow for a fair comparison.

In the annotations section (see Section 3.2), we found that most ironic tweets in our corpus (i.e., 72%) show a contrast between a positive and a negative polarity expression. In the following paragraphs, we aim to verify whether this category is also the most likely to be recognized automatically, as compared to other irony types. To verify the validity of our assumption, we analyzed the classification output for the different irony types in our corpus. Figure 2 visualizes the accuracy of the best-performing classifier (i.e., lexical + semantic + syntactic features) for each irony type and the different types of non-ironic tweets (i.e., hashtag vs. background corpus). The bar chart seems to confirm our intuition that the system performs best on detecting ironic tweets that are realized by means of a polarity contrast (78% accuracy), followed by instances describing situational irony. On the other hand, detecting *other type of irony* appears much more challenging (45%). A closer look at *other verbal irony* reveals that the instances are often ambiguous and realized in diverse ways, as shown in Examples (10) and (11). It is important to recall that, prior to classification, the hashtags *#irony*, *#sarcasm*, and *#not* were removed from the tweets.

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

(11) 'I like to think of myself as a broken down Justin Bieber' – my philosophy professor, everyone #sarcasm

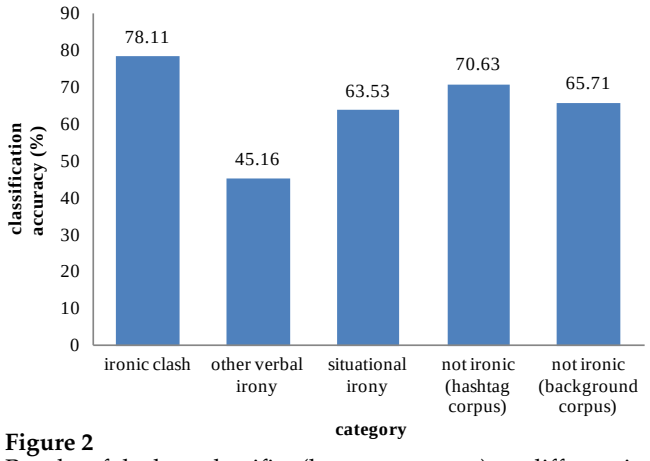


Figure 2
Results of the best classifier (lex + sem + synt) on different irony types.

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. We see two possible explanations for this. First, we observed that in the majority (77%) of the misclassified tweets, the only clue for a polarity contrast was an irony-related hashtag (i.e., "#not"), which was removed from the data prior to training. In fact, as illustrated by Example (12), without such a meta-hashtag, it is very difficult to know whether the instance is ironic.

(12) Thanks dad for your support! #not

Second, tweets that do not require a meta-hashtag to perceive a polarity contrast, but that were nevertheless missed by the classifier (23%), included an evaluation as part of a hashtag (e.g., "#thisonlygetsbetter") or an **implicit evaluation** (Example (13)). As explained in the Introduction, understanding such implicit sentiment requires connotative (world) knowledge.

(13) Spending the majority of my day in and out of the doctor$^{[NEG]}$ has been awesome. #sarcasm

Whereas the polarity opposition in Example (12) would be impossible—even for humans—to recognize without hashtag information, the polarity contrast in Example (13) is likely to be identified, on the condition that the system could access common sense or connotative knowledge. As such, it would ideally recognize phrases like "spending (...) day in and out of the doctor" as related to negative sentiment, and find the contrast with the positive expression "awesome."

In the next section, we therefore take a closer look at the implicit sentiment expressions (or **targets**) that were annotated in the irony corpus and take the first steps to detect such implicit or prototypical sentiment automatically.

4. A Hybrid Approach to Model Implicit Sentiment

In the previous section, we observed that the majority of ironic tweets show a polarity contrast between what is said and what is implied, or, more specifically, a literal positive evaluation that is contrasted with an implicit negative evaluation, or vice versa. To recognize the irony in such tweets, it is key to identify the words that realize this polarity contrast. Although explicit sentiment expressions are mostly traceable using a lexicon-based approach, a bigger challenge 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.

In this section, we confront the challenge of automatically recognizing the implicit sentiment related to particular concepts or situations. As explained in Section 2, Riloff et al. (2013) took a bootstrapping approach to learn negative situation phrases (i.e., verbs) in the vicinity of positive seed words. Having at our disposal manually annotated implicit sentiment phrases[10] (e.g., "spending the majority of my day in and out of the doctor," "working in the weekend"), our goal is to develop a method to define the implicit sentiment related to these targets automatically. We propose two methods to tackle this problem: (i) based on SenticNet, an existing knowledge and *sentics* database, and (ii) through a data-driven approach using Twitter. Both methods will be evaluated against the gold-standard annotations.

10 Situations and concepts that carry prototypical sentiment are called **targets** in the annotation scheme.

Table 8
Excerpt of the manually annotated implicit sentiment phrases or **targets**.

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
130km #cycle tomorrow, in the minus degree weather	negative

Table 8 presents a number of example targets in our corpus and the gold-standard implicit sentiment related to them. In total, 671 unique targets were annotated, 665 of which have a negative connotation, and 6 of which have a positive connotation. This imbalance between positive and negative targets confirms earlier findings that irony is more frequently realized by saying something positive while meaning something negative than the other way around (Riloff et al. 2013; Van Hee, Lefever, and Hoste 2016c).

During the annotation procedure, annotators were tasked with assigning the prototypical sentiment (i.e., the feeling a concept generally invokes for a group of people) related to each of the concepts. Whether a concept invokes a positive or negative sentiment is a subjective judgment defined by personal or cultural differences. Throughout the annotations, the annotators were therefore asked to take the rest of the tweet into account as the context to get an impression of the intended sentiment, and to judge as generally as possible by prioritizing commonly held opinions over their own. For instance, although some people may like, or do not mind, to work on festive days, "working on Christmas" was attributed a negative connotation, assuming that the majority of people would not like it. An inter-annotator experiment confirmed that, despite the subjective nature of the task, fairly good agreement (κ = 0.66 [round 1] and κ = 0.55 [round 2]) was obtained. The manual annotations of implicit sentiment as illustrated here will serve as the gold standard against which we will compare our methodology to infer implicit sentiment automatically (later in this article).

4.1 Using SenticNet to Infer Implicit Sentiment

Our first approach to define the implicit sentiment of the targets is a knowledge base approach by 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 real-world concepts, instantiated by single words and multiword expressions, such as "miss flight," and "celebrate special occasion." SenticNet was not built by manual labeling of existing resources, but is automatically generated via graph-mining and dimensionality reduction techniques applied to multiple commonsense knowledge sources (Cambria et al. 2010).

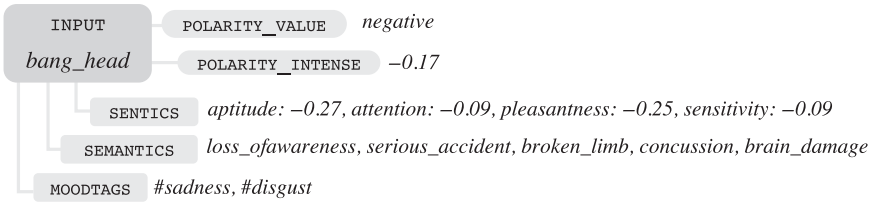


Figure 3
Example of the SenticNet output for the entry "bang_head."

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 2013). Its ability to represent polarity values for natural language concepts (e.g., "exam," "lose temper") allows it to outperform other sentiment resources like 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. A SenticNet example is presented in Figure 3.

The knowledge base contains mainly unigrams, bigrams, and trigrams, so most targets contain more words than would fit in a single query to the database (see Table 8). Consequently, in the case of a target being a multiword expression or a phrase, the overall polarity had to be defined based on the polarities of the individual words in the target. The following paragraphs zoom in on our approach to implicit sentiment modeling using SenticNet 4 (Cambria et al. 2016). 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).

We defined the implicit sentiment of each target by looking up (i) all words in the target, (ii) only content words, and (iii) multiwords. As content words, we considered nouns, adjectives, adverbs, and verbs, based on the part-of-speech output of the LeTs Preprocess toolkit (Van de Kauter et al. 2013). This way, polarity values for function words like prepositions and numerals were not taken into account. For multiword look-up, we used Rajagopal et al.'s (2013) concept parser, which makes use of SenticNet as its knowledge base and decomposes the input phrase into commonsense concepts contained by the knowledge base. Figure 4 visualizes such a multiword look-up by means of a flowchart depicting the process from input query to the overall target polarity.

Prior to the actual look-up, a number of preprocessing steps were undertaken. First, URLs and @-replies were discarded because they have no coverage (i.e., they are not present) in SenticNet. For the same reason, hash signs ("#") were stripped from hashtag words and punctuation marks were removed. Second, concatenated words were split based on casing (e.g., "noThanks" → "no thanks"). Third, common abbreviations were replaced based on an existing dictionary,[11] and contractions were expanded

11 Source: `https://slangit.com`.

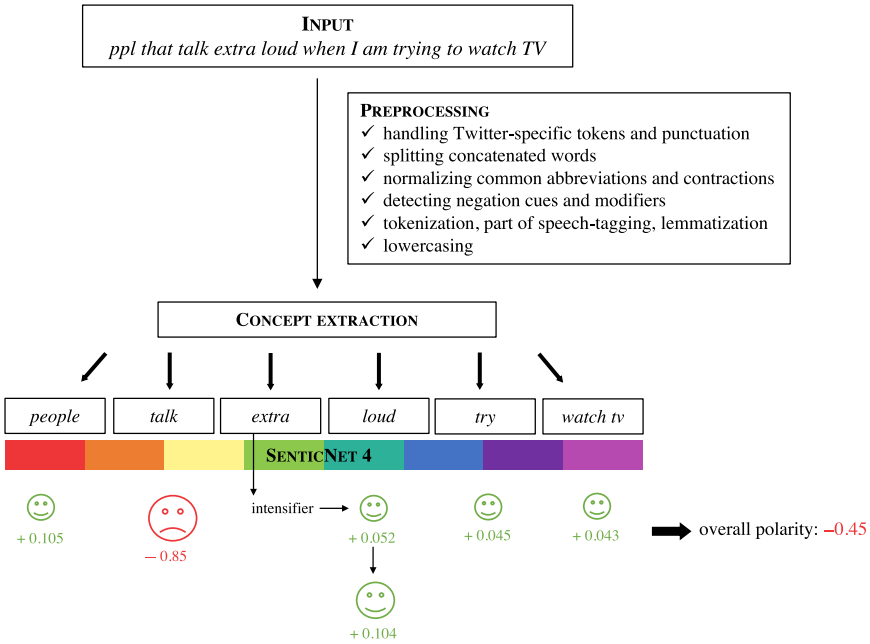


Figure 4
Flowchart visualizing concept look-up using SenticNet 4.

(e.g., "wouldn't" → would not), because SenticNet contains only full forms. In a next step, negation words (Example (14)) and modifiers (Example (15)) were identified.

(14) not[NEG] getting any sleep

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

If a sentiment word was preceded by a negation word, its polarity value was inverted. When preceded by a modifier, its polarity was increased (*2) or decreased (*0.5), depending on the modifier type (e.g., intensifier or diminisher). Next, all targets were tokenized, part-of-speech-tagged, and lemmatized using LeTs preprocess (Van de Kauter et al. 2013) so that lemmas rather than words could be considered for look-up. In a final preprocessing step, SenticNet queries were lowercased.

Figure 4 visualizes the automatic sentiment-determining process, starting with the targets as input queries for SenticNet. The queries were preprocessed and broken down into single words or concepts, which were looked up in the knowledge base. SenticNet polarities for words or concepts were then summed to generate an overall sentiment value for the target.

4.1.1 Analysis. To evaluate our approach, we compared the SenticNet polarities for each target with the gold-standard annotations. Table 9 presents the accuracy of our SenticNet-based polarity assignment by respectively looking at all words, only content words, and multiword expressions in the target. Table 9 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

Table 9
Automatically assigned implicit sentiment using SenticNet 4.

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

account polarity values related to function words, which sometimes has a positive effect on the total target polarity (Example (16)).

(16) **target:** *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 it protects some "semantic atoms" (Cambria and Hussain 2015) that lose their original meaning when broken down into single words. 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. For instance, although "December" may evoke a positive sentiment for many people, when combined with "icy roads" or "electric bills," it becomes negative. Such very specific multiword terms are, however, not contained in SenticNet. The overall SenticNet polarity of this example would be positive (*december* (+0.799) + *electric_bill* (−0.04)), although most people would probably agree on the concept's negative connotation.

Other challenges when using SenticNet to assign polarities to concepts are (i) the lack of coverage for some words (e.g., "rancid"), and (ii) the limited number of inflected forms in the database.

Although there is room for optimization 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 (Example (17)).

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

In sum, although knowledge bases like SenticNet present a convenient resource for word-level sentiment analysis, a more complex approach would be required to define the implicit sentiment of phrases, which often require reasoning or context interpreting. This involves knowing that people do not like working a double shift, especially not on holidays. Still, such a knowledge base would suffer the drawback of its static nature, because even when containing a massive amount of information, it could probably not keep pace with the rapidly evolving world around us, causing common sense to be continuously updated.

4.2 Crawling Twitter to Infer Implicit Sentiment

In the previous section, we used SenticNet 4 to infer implicit sentiment related to the targets and revealed a number of drawbacks, among others the inability to define the sentiment of phrases without decomposing them. In this section, we take a machine learning approach to define implicit sentiment based on crawled tweets. We verify the hypothesis that Twitter provides insights into connotative 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. Moreover, the medium allows us to collect real-time opinions held by a large group of people, whereas knowledge bases are generally static and rely on knowledge that has been derived automatically or inserted by a restricted number of experts.

4.2.1 Method. An important first step to infer implicit sentiment using Twitter is collecting sufficient tweets for each target so that a reliable estimation can be made of its prototypical sentiment. We recall that targets are phrases that describe 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 and subsequently determined the prevailing sentiment in these tweets using supervised machine learning (Van Hee et al. 2014).

To this purpose, we applied a sentiment classifier the architecture of which is described in Van Hee et al. (2017). The system is trained on data distributed in the framework of the SemEval-2014 shared task on *Sentiment Analysis in Twitter* (Rosenthal et al. 2014) and optimized through feature selection and hyperparameter estimation. The classifier predicts the overall polarity of a tweet as positive, negative, or 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 the polarity 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 like tweets would provide insights into the typical sentiment that a concept evokes, or its connotation. For instance, when a large group of people complain about attending lectures at 8 a.m., one could assume it is generally considered unpleasant. We started with the originally annotated targets as Twitter queries, but explored a number of abstraction methods to see whether they are likely to improve the coverage (see further).

4.2.2 Original Annotations as Twitter Queries. Each target was used as a Twitter search query. As a first step, all targets were preprocessed to make look-up as effective as possible. We briefly describe the preprocessing steps, since they are similar to the SenticNet approach (see Section 4.1) and include (i) handling of Twitter-specific tokens (i.e., URLs and @-replies were removed and hash-signs were stripped off), (ii) removal of punctuation marks, (iii) splitting of concatenated words based on casing, (iv) lowercasing, and (v) replacement of ampersands, as they have a syntactic function in a Twitter search query.

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 irony-related hashtags, since we aim to get insights into sincere (i.e., non-ironic) opinions and sentiment related to the targets.

Once a set of tweets was collected and cleaned for each target, we used our sentiment analysis pipeline to define per tweet whether it was positive or negative. We then 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 negative, the prototypical sentiment related to this situation was defined as negative. The automatically defined implicit sentiment values were then evaluated against the gold-standard labels from the manual annotations (see Section 3.2).

Figure 5 visualizes 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 (239 out of 671) when looked up in their original form. A possible explanation for the limited coverage is that many targets were too specific to yield many tweets as they contained digits (Example (18)), personal pronouns, or were rather long (Example (19)). In fact, analysis revealed that the average length of targets for which at least one tweet had been found was three tokens, whereas it was nine for targets yielding no tweets.

(18) 7:30 finals on a friday

(19) when someone accidentally deletes everything on your phone

Another explanation for the limited coverage of the targets is methodology-related. Using the Twitter Search API does not allow one to retrieve historical tweets, but only returns tweets that match the input query from the past 7 days. As a consequence, some targets yielded very few or even no results, and, obviously, the longer and more specific the target, the fewer tweets were found.

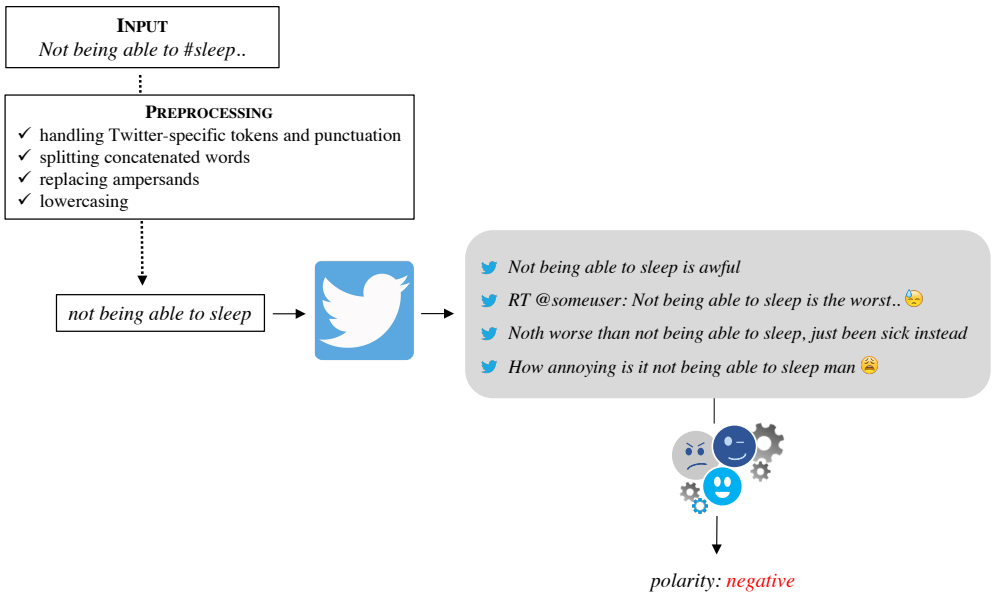


Figure 5
Defining the implicit sentiment of a target using Twitter.

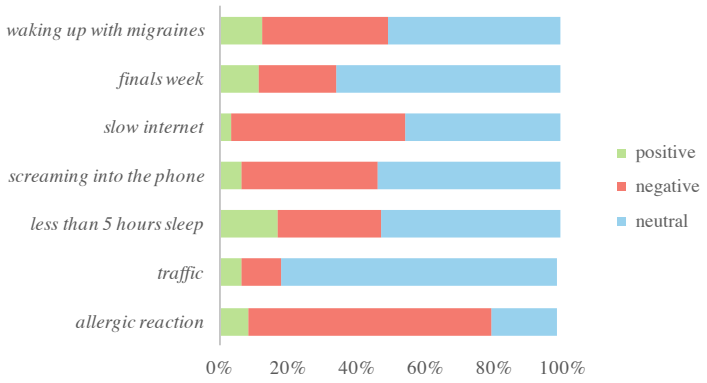


Figure 6
Proportion of positive (green), negative (red), and neutral (blue) tweets for a set of example targets.

As mentioned earlier, after collecting a number of tweets for a target, automatic sentiment analysis was applied to determine the general sentiment expressed toward the target. Figure 6 visualizes the sentiment analysis output for some 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 toward "allergic reaction" and "slow internet" were mostly negative, whereas they were mostly neutral for "traffic" (i.e., tweets that communicate objective traffic information or that refer to internet traffic, like "Ignite Your #Blog Traffic With #Content-Marketing `https://t.co/xt6wU4JfkY`"). The blue bars represent neutral tweets that contain either no sentiment, or both positive and negative sentiments. Because we aim to infer connotative information, such tweets are less informative for this task and we therefore defined the overall polarity of a concept with and without considering neutral tweets.

Table 10 shows the coverage of the original targets (cf. Table 8) on Twitter and presents the accuracy of our method to define their implicit sentiment. Considering all tweet predictions for a given target resulted in a low accuracy because many tweets were neutral. If we take a look at Examples (20) and (21), we observe that negative concepts (e.g., "inflation climbs," "people who lie") may occur in neutral tweets (Example (20)) or tweets expressing both a positive and negative sentiment (Example (21)).

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

(21) 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?

Table 10
Sentiment analysis accuracy after crawling Twitter using the original targets.

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

Considering the most prevalent sentiment after discarding such neutral tweets resulted in much better accuracy—72%. This means that for 72% of the targets, we were able to define their implicit sentiment. This also means, however, that for 28% of the targets the predicted sentiment was incorrect. A qualitative analysis indicated several reasons for this: (i) not all tweets mentioning a negative target were actually negative; and (ii) tweets were sometimes misclassified due to ambiguity.

In sum, Table 10 confirms our hypothesis that Twitter data offer insights into the prototypical sentiment related to particular concepts or situations. It is important to underline, however, that our 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 8, 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.

4.2.3 Content Words as Twitter Queries. First, we reduced the targets to content words only. Based on part-of-speech information obtained using LeTs Preprocess (Van de Kauter et al. 2013), 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 look-up.

As shown in Table 11, keeping only content words discards pronouns, determiners, and so forth, and makes 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, when keeping only content words, the target "9 am lectures" becomes "*lectures," which could generate a number of irrelevant tweets when used as a search query. Overall, using content words instead of the original targets provides some abstraction, allowing one to collect tweets for 277 out of the 671 targets. This is 5% more than when using the original targets as queries. On the downside, the likelihood of retrieving irrelevant tweets increases. Here are two example tweets that correspond to the query "* hour car ride," derived from the target "10 hour car ride." When comparing

Table 11
Original targets versus content word targets. Function words are replaced by wildcards.

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 12
Sentiment analysis accuracy based on a Twitter crawl using content word targets.

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

Examples (22) and (23), we see that even numerals can be essential for the semantics of a phrase, and hence to its connotation.

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

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

Table 12 shows the coverage and sentiment analysis results for the targets based on content words, again before and after discarding neutral tweets. Although wildcards in the search query are prone to increase the number of irrelevant tweets, 72.20% of the targets were assigned the correct implicit sentiment, which is slightly more compared with the original targets approach (cf. Table 10).

4.2.4 Dependency Heads as Queries. As a second method to make abstraction from the original targets, we made use of dependency relations within each target. We considered the head of a dependency relation in a target, 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) that modify it. We made use of the statistical dependency parser implemented in the Python library spaCy,[12] as it has shown to achieve a state-of-the-art performance (Choi, Tetreault, and Stent 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 from 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 13 presents some example targets for which dependency heads were extracted. Similarly to the content-words approach (cf. Table 11), words that had been discarded were replaced by wildcards ("*"). As shown in Table 14, using dependency heads rather than the original targets allowed us to collect tweets for 347 out of the 671 targets (52%). 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%.

4.2.5 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

12 http://spacy.io.

Table 13
Original targets versus dependency heads in the targets.

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
I have pink eye	* have * eye
when someone accidentally deletes everything on your phone	* * * deletes everything on * phone
9 am lectures	* am lectures

Table 14
Sentiment analysis accuracy based on a Twitter crawl using dependency heads as queries.

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

are typical structures for negative situation phrases that are common in ironic tweets. Table 15 presents some example targets and the VO patterns that were extracted. No such patterns could be derived for noun phrase targets, which are indicated with "n.a." in the table.

Table 16 presents the coverage (i.e., the proportion of targets for which we were able to retrieve tweets) and sentiment analysis results obtained using VO sequences in our targets as queries. If more than one VO phrase had been extracted from a target, we considered the predicted sentiment of all phrases in that target (i.e., "positive" if all VO strings were positive, "negative" if all were negative, "neutral" if some were positive and others negative). The method allowed us to collect tweets for 312 out of the 671 targets (47%), which is more than that obtained with the original targets and content word targets (because VO patterns are more general than the original or content word targets), but fewer than the dependency heads approach. Sentiment analysis performance is slightly lower compared with the other approaches. An explanation is that extracting VO patterns from concepts or situation phrases implies a greater loss of information. For instance, reducing the phrase "taking the subway alone at 2:40 a.m." to "taking subway" discards the element that invokes the negative sentiment (i.e., "alone at 2:40 a.m."). In other examples, keeping only VO patterns implies that the implicit sentiment of the original target becomes less strong (e.g., "work a double on New Year's Eve" $\rightarrow$ "work double"). Also, Table 15 suggests that considering VO 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.

4.2.6 Analysis. In the previous paragraphs, we explored four methods to crawl tweets for a set of connoted situations (or targets) that were manually annotated. We can

Table 15
Verb-object patterns of targets.

original target	VO 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 16
Sentiment analysis accuracy based on a Twitter crawl using dependency heads as queries.

targets	coverage	accuracy (pos/neg/neu)	accuracy (pos/neg)
VO patterns	46.50%	17.68%	68.17%

conclude that applying sentiment analysis to the tweets is a viable method to define the prototypical sentiment related to the situations (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 fewer than half of the targets. Analysis revealed that although some information can be discarded without meaning loss (e.g., pronouns, determiners), removing other elements (e.g., numerals) does imply a change in meaning (e.g., "10 hour car drive" versus "hour car drive").

For practical motivations, we defined a maximum of 500 search results when crawling tweets. Given that most targets were very specific, and the API restrictions imply that no historical results can be returned, most targets did not even reach this maximum. We wanted to investigate, however, whether sentiment accuracy increases with the number of tweets returned for a target. One could hypothesize 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 its prototypical sentiment.

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 7.

As can be inferred from Figure 7, collecting more tweets seems to have a moderate effect on the overall sentiment analysis performance (91% vs. 94%). However, the increase seems to stagnate at 750 tweets and the scores even decline as the number of tweets further increases. This would suggest that using more tweets to determine

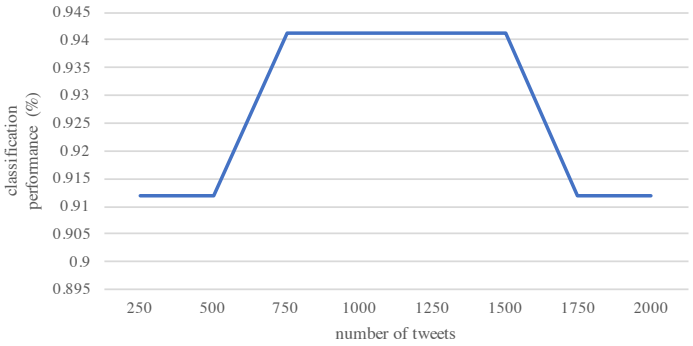


Figure 7
Sentiment analysis results by gradually incrementing the number of tweets.

the prototypical sentiment related to a concept does not necessarily provide a better indication of that sentiment. One reason could be that collecting more tweets could result in more irrelevant tweets to be retrieved. The effect we measured here is based on a very small sample (34 targets), so the results should be interpreted carefully.

In sum, an important advantage of using Twitter to infer connotative knowledge, as compared with SenticNet, is that it allows us to look up phrases without needing to decompose them 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, first, that it is more complex than a knowledge base look-up 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 these tweets. Second, the prototypical sentiment of a situation being based on real-time opinions, it might be influenced by crises or trends, which may cause fluctuations in the public opinion toward a specific concept or situation. Depending on what one is looking for (i.e., implicit sentiment of a concept at a particular point in time, or over a longer period), this can be an advantage or disadvantage.

5. Using Implicit Sentiment for Irony Detection

In the previous section, we showed that analyzing opinions expressed by the "Twitter crowd" is a good strategy to infer implicit sentiment or connotative knowledge related to specific concepts or situations. In this section, we combine this method with the identification of explicit subjective words to define whether a polarity contrast is present in a tweet and add this information to our SVM classifier. The reported results are obtained on the held-out test data.

First, it should be noted that a contrast feature (based on explicit polarity words) was already included as part of the sentiment lexicon features (see Section 3). However, the feature is not included in the final irony classifier, as the experiments revealed that combining lexical, semantic, and syntactic features without sentiment features works best. In this section, we further elaborate this polarity contrast feature by considering both explicit and implicit polarities and re-evaluate our SVM classifier when informed with such polarity contrast information. This is done in two ways:

1. By means of a **binary feature** indicating the presence of a polarity contrast (i.e., between either explicit or explicit and implicit polarities) in a tweet;

2. As a class label for irony (i.e., if a polarity contrast is present, the tweet is ironic, otherwise it is not). This prediction is combined with the SVM prediction into a **hybrid system**.

The hybrid system is applied in two flavors: A tweet is predicted as ironic if (i) both the SVM and clash-based system consider it ironic and (ii) one of the two systems predicts it as ironic.

5.1 Identifying a Polarity Contrast

The polarity contrast approach to irony detection is implemented in different ways. First, in ironic tweets, an explicit evaluation can be contrasted with (i) another explicit evaluation (it is denoted as [exp-exp] in the tables) or (ii) an implied evaluation ([exp-imp] in the tables). We evaluate the system's performance on detecting both types of polarity contrasts.

We used the Twitter-based method as described in Section 4.2.3 to define the implicit sentiment related to so-called targets (see Section 3.2) and made use of sentiment lexicons (cf. Section 3.5) to identify explicit sentiment expressions. The system is evaluated in two ways: (A) by providing it with gold-standard implicit sentiment and (B) by determining the implicit sentiment of the targets automatically. We illustrate the approach with Example (24).

(24) I just love when you test my patience! 😌

Evaluation A implies that the system was provided with the target "you test my patience" and its negative implicit polarity. It subsequently searched for a contrastive polarity word in the remainder of the tweet using sentiment lexicons (e.g., "love"). Evaluation B implies that the system was provided with the target "you test my patience" and automatically defined its implicit sentiment. Like in A, it then searched for contrastive polarity words in the remainder of the tweet.

Table 17 presents the scores of the system on the test data (cf. Section 3.2). Recall, precision, and F_1 score are calculated on the positive class (i.e., ironic) instances. 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 Section 3 (F_1 = 70.11%). This can be explained by the fact that the contrast system is targeted toward instances where the irony results from a polarity contrast, which "merely" constitute 70% of the irony class (see earlier). Moreover, analysis of the manual annotations revealed that, in about 50% of the ironic instances, an irony-related hashtag is required to infer the irony, as shown in the following example.

(25) My english.. soo perf in the morning ✌ #not

Given that irony-related hashtags like *#not* are removed from the tweets, such instances cannot be detected by the polarity contrast system, whereas the SVM classifier might pick up other indicators of irony (e.g., punctuation, flooding). Our qualitative analysis further revealed that the systems in set-up 1 and 3 tend to overgenerate, as they also predict an instance as ironic if contrasting polarity words are found in a tweet, even if it is not ironic (Example (26)).

Table 17
Performance of the clash-based system for irony detection using gold-standard and automatic implicit sentiment information.

	positive class	clash	implicit sentiment	accuracy	precision	recall	F_1
1	ironic by clash + situational + other	[imp-exp] or [exp-exp]	gold standard (A)	56.99%	57.78%	55.88%	56.81%
2	ironic by clash + situational + other	[imp-exp]	gold standard (A)	61.80%	100%	24.54%	39.40%
3	ironic by clash + situational + other	[imp-exp] or [exp-exp]	automatic (B)	51.88%	52.86%	45.77%	49.06%
4	ironic by clash + situational + other	[imp-exp]	automatic (B)	55.11%	100%	11.34%	20.37%

(26) work hard[NEG] in silence ; let success[POS] make the noise[NEG].

Although the clash-based system does not outperform the SVM classifier, it is worthwhile to note that the former is able to recognize ironic instances (58 to be precise) that the SVM classifier overlooks, including Examples (27) and (28).

(27) Spending the majority of my day in and out of the doctor[NEG] has been awesome[POS].

(28) Literally half of the finals i have this semester are today[NEG] , and that's totally not stressful[POS] at all!

Moreover, being able to recognize implicit–explicit contrasts, the system achieves maximum precision.

5.2 Irony Detection Based on (Implicit) Polarity Contrasts

In a next step, we used the polarity contrast system as described in the previous paragraph to inform our SVM-based irony classifier. As explained in the introduction of this section, we combined the information provided by the two systems in two ways. First, we included the output of the contrast-based method as a binary feature for the SVM classifier. 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 (see Table 18). We considered the original SVM classifier as the baseline. As can be deduced from the table, adding a polarity contrast feature only caused a slight performance improvement if gold-standard implicit information was used. Precision went up by 1.3 points, but recall of the system decreased by 1.2 points. When using automatically derived implicit sentiment, the system scored equally compared to the baseline, hence the feature does not seem to add crucial information to the model. As was also observed in the qualitative analysis described in Section 3.8.4, classification errors on the *ironic by clash* category mainly include tweets where a polarity contrast is difficult to perceive, even if implicit sentiment is taken into account. Examples are tweets where the only clue for a polarity contrast was an irony-related hashtag (Examples (25) and (12)) and tweets where the contrasting polarity is difficult to grasp, for instance when in a concatenated hashtag (e.g., "#thisonlygetsbetter").

Table 18
Performance of the SVM+clash system for irony detection using gold-standard and automatic implicit sentiment information.

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%

Furthermore, it is worth noting that the baseline exploits a rich and optimized feature set and 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, and individual feature selection will be investigated in the future to gain better insights into the system performance.

In a second set of experiments, we implemented a hybrid system for irony detection. Two conditions were tested for the system to define whether an instance was ironic: (i) both systems predicted the tweet as ironic (*AND-combination*), and (ii) one of the two systems predicted the tweet as ironic (*OR-combination*). Table 19 presents the results of this hybrid irony detection system. The table reveals that, when looking at the combined set-ups, systems 1 and 2 outperform 3 and 4, as the former rely on gold-standard implicit sentiment information. Whereas the baseline scores best in terms of F_1 score, systems 2 and 4 yield much higher recall, showing that both approaches are complementary, and system 1 and 3 outperform the baseline in terms of precision. This makes logical sense, because 1 and 3 require both systems to predict an instance as ironic for the final prediction. When comparing the results with Table 18, we observe that depending on how the two systems are combined (i.e., AND/OR), precision and recall are substantially better than when including polarity contrast as a feature. This demonstrates that polarity contrast information has a strong potential to improve irony detection, on the condition that the information provided by both systems (i.e., SVM and polarity contrast) is properly combined.

Finally, given that the contrast-based method is targeted toward instances of irony that include a polarity clash, we calculated its performance on this specific irony type in the test corpus. This confirmed the validity of the contrast-based approach as we observed that, when combining the original SVM with a polarity contrast system (OR-combination), we were able to recognize 96% of the *ironic by clash* instances if no hashtag was required to infer the irony.

Overall, the results demonstrate that our approach compares favorably with that of Riloff et al. (2013), who applied bootstrapped learning to extract positive sentiment and negative situation phrases from hashtag-labeled ironic tweets. Their combined method (contrast-based system + SVM classifier) yielded an F-score of 51% and recall of 44%. Moreover, whereas their approach requires a large irony corpus to extract implicit sentiment phrases, we were able to recognize implicit sentiment based on real-time Twitter data, without requiring any other training data than the annotated targets. As opposed to the researchers, however, we did not address the problem of identifying implicit sentiment phrases in text, but started from manually annotated targets. Identifying

such targets automatically in tweets will, however, be an interesting direction for future research.

6. Conclusions

This article set out to explore automatic irony detection on Twitter by making use of implicit sentiment. We present, to our knowledge, the first approach to include explicit and implicit polarity contrast information for irony detection based on prototypical sentiment that is automatically extracted from real-time tweets.

We developed an SVM-based irony detection system exploiting lexical, syntactic, and semantic features that is trained on a manually annotated irony corpus. Similar features are commonly used in irony literature, but we expanded our lexical and semantic feature sets with, respectively, language model features and word cluster information—two features that have, to our knowledge, not been sufficiently explored for this task. Using these feature groups, a series of binary classification experiments were carried out and evaluated against three baselines. Using a combined feature set, our classifier yielded an F_1 score of 70.11% and outperformed the strong character *n*-gram baseline. While our experiments describe manual feature group selection, weighting and selecting individual features will be a crucial direction for future work to optimize the classifier by removing redundant information and gaining more insights into the most contributive features for irony detection.

We found that the classifier performs best on ironic instances with a polarity contrast, although this category presents an important challenge, being implicit or prototypical sentiment related to particular situations (e.g., "going to the dentist"), which cannot be captured using traditional sentiment lexicons. We therefore investigated how such implicit sentiment can automatically be inferred. Starting from manually annotated connoted situations (targets), we determined their prototypical sentiment using SenticNet 4 and real-time crawled tweets. The experiments revealed that applying sentiment analysis to a set of tweets about a concept or situation is a viable method to determine the implicit sentiment related to that concept or situation. Knowledge bases like SenticNet, by contrast, are often restricted by their limited coverage, static character, and tendency to contain mainly individual words instead of concepts or phrases. Being able to infer implicit sentiment automatically with real-time tweets (accuracy = 72%), we

Table 19
Performance of the hybrid approach to detecting irony using automatic and gold-standard implicit sentiment information.

	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%

informed our irony detection system with explicit and implicit polarity contrast information and observed a high recall of the system, especially on *ironic by clash* instances.

As we started from manually annotated targets, an important direction for future work will be to identify such targets (or connoted situations) automatically in tweets, namely, in which text spans are they realized? Furthermore, while the annotation guidelines distinguish between different irony types, the present research approaches irony detection as a binary classification task and hence provides insights into the feasibility of irony detection in general. However, fine-grained irony classification might be worthwhile in the future, to be able to detect specifically ironic tweets in which a polarity inversion takes place.

Finally, we collected our irony corpus with the hashtags #*irony*, #*sarcasm*, and #*not* and manually annotated different irony types irrespective of the hashtags present in a particular tweet. As was shown by Sulis et al. (2016), different irony hashtags are likely to indicate different types of irony, which are easier or more difficult to detect by a classifier. Exploring the performance of our classifier on tweets with the hashtag #*irony* versus tweets with #*sarcasm* and #*not* will therefore constitute an interesting research direction in the future.

References

Balahur, Alexandra, Jesús M. Hermida, Andrés Montoyo, and Rafael Muñoz. 2011. EmotiNet: A knowledge base for emotion detection in text built on the appraisal theories. In *Proceedings of the 16th International Conference on Natural Language Processing and Information Systems (NLDB'11)*, pages 27–39, Heidelberg.

Balahur, Alexandra and Hristo Tanev. 2016. Detecting implicit expressions of affect from text using semantic knowledge on common concept properties. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1165–1170, Paris.

Bamman, David and Noah A. Smith. 2015. Contextualized sarcasm detection on Twitter. In *Proceedings of the Ninth International Conference on Web and Social Media (ICWSM)*, pages 574–577, Oxford.

Bojar, Ondrej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation (WMT16). In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 131–198, Berlin.

Bosco, Cristina, Mirko Lai, Viviana Patti, and Daniela Virone. 2016. Tweeting and being ironic in the debate about a political reform: The French annotated corpus Twitter-MariagePourTous. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož.

Bouazizi, Mondher and Tomoaki Ohtsuki. 2016. Sarcasm detection in Twitter: "All your products are incredibly amazing!!!" - are they really? In *2015 IEEE Global Communications Conference, GLOBECOM 2015*, pages 1–6, San Diego, CA.

Burgers, Christian. 2010. *Verbal Irony: Use and Effects in Written Discourse.* Ph.D. thesis, UB Nijmegen [Host].

Buschmeier, Konstantin, Philipp Cimiano, and Roman Klinger. 2014. An impact analysis of features in a classification approach to irony detection in product reviews. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 42–49, Baltimore, MD.

Cambria, Erik, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco, editors. 2017. *A Practical Guide to Sentiment Analysis*. Springer International Publishing, Cham, Switzerland.

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

Cambria, Erik, Amir Hussain, Catherine Havasi, and Chris Eckl. 2009. Common

sense computing: From the society of mind to digital intuition and beyond. In Julian Fierrez, Javier Ortega-Garcia, Anna Esposito, Andrzej Drygajlo, and Marcos Faundez-Zanuy, editors, *Biometric ID Management and Multimodal Communication: Joint COST 2101 and 2102 International Conference, BioID_MultiComm 2009*, Springer, Berlin, Heidelberg, Madrid, pages 252–259.

Cambria, Erik, Soujanya Poria, Rajiv Bajpai, and Bjoern Schuller. 2016. SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, pages 2666–2677, Osaka.

Cambria, Erik, Robert Speer, Catherine Havasi, and Amir Hussain. 2010. SenticNet: A publicly available semantic resource for opinion mining. In *AAAI Fall Symposium: Commonsense Knowledge (AAAI Technical Report)*, volume FS-10-02, pages 14–18.

Camp, Elisabeth. 2012. Sarcasm, pretense, and the semantics/pragmatics distinction. *Nous*, 46(4):587–634.

Carletta, Jean. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics.*, 22(2):249–254.

Chang, Chih-Chung and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27:1–27:27.

Choi, Jinho D., Joel R. Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using A web-based evaluation tool. In *ACL (1)*, pages 387–396.

Davidov, Dmitry, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL'10)*, pages 107–116, Uppsala.

Deng, Lingjia and Janyce Wiebe. 2014. Sentiment propagation via implicature constraints. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL'14)*, pages 377–385, Gothenburg.

Deriu, Jan, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. SwissCheese at SemEval-2016 Task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1124–1128, San Diego, CA.

Dinakar, Karthik, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems*, 2(3):18:1–18:30.

Ebrahimi, Monireh. 2013. Side effects recognition as implicit opinion words in drug reviews. Master's thesis, University Teknologi Malaysia.

Farías, Delia Irazú Hernańdez, Viviana Patti, and Paolo Rosso. 2016. Irony detection in Twitter: The role of affective content. *ACM Transactions on Internet Technology*, 16(3):19:1–19:24.

Feng, Song, Jun Sak Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1774–1784, Sofia.

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

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

Ghosh, Aniruddha and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, CA.

Gimpel, Kevin, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 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)*, pages 42–47, Portland, OR.

González-Ibáñez, Roberto, Smaranda Muresan, and Nina Wacholder. 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)*, pages 581–586, Portland, OR.

Greene, S., Charles. 2007. *Spin: Lexical Semantics, Transitivity, and the Identification of Implicit Sentiment*. Ph.D. thesis, University of Maryland.

Greene, Stephan and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '09)*, pages 503–511, Stroudsburg, PA.

Grice, H. Paul. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics: Vol. 3*. Academic Press, New York, pages 41–58.

Grice, H. Paul. 1978. Further notes on logic and conversation. In Peter Cole, editor, *Syntax and Semantics: Vol. 9*. Academic Press, New York, pages 113–127.

Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia.

Hogenboom, Alexander, Danella Bal, Flavius Frasincar, Malissa Bal, Franciska De Jong, and Uzay Kaymak. 2015. Exploiting emoticons in polarity classification of text. *Journal of Web Engineering*, 14(1-2):22–40.

Hoste, Véronique, Els Lefever, Stephan van der Waart van Gulik, and Bart Desmet. 2016. TripleSent: A triple store of events associated with their prototypical sentiment. In *Proceedings of the Eighth International Conference on Information, Process, and Knowledge Management*, pages 91–93, Venice.

Hsu, Chih Wei, Chih-Chung Chang, and Chih-Jen Lin. 2003. A practical guide to support vector classification, Technical report, Department of Computer Science, National Taiwan University. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

Joshi, Aditya, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, 50(5):73:1–73:22.

Joshi, Aditya, Vinita Sharma, and Pushpak Bhattacharyya. 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 Natural Language Processing (Short Papers)*, pages 757–762, Beijing.

Joshi, Aditya, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark J. Carman. 2016. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, pages 1006–1011, Austin, TX.

Karoui, Jihen, Farah Benamara, Véronique Moriceau, Nathalie Aussenac-Gilles, and Lamia Hadrich Belguith. 2015. Towards a contextual pragmatic model to detect irony in tweets. In *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*, pages 644–650, Beijing.

Karoui, Jihen, Farah Benamara, Véronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. 2017. Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. In *EACL-European Chapter of the Association for Computational Linguistics*, pages 262–272, Valencia.

Keerthi, Sathiya S. and Chih-Jen Lin. 2003. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation*, 15(7):1667–1689.

Khattri, Anupam, Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2015. Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA'15)*, pages 25–30, Lisbon.

Kralj Novak, Petra, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of emojis. *PLOS ONE*, 10(12):1–22.

Kunneman, Florian, Christine Liebrecht, Margot van Mulken, and Antal van den Bosch. 2015. Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management*, 51(4):500–509.

Landis, J. Richard and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Lehmann, Jens, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

Lenat, Douglas B. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Liebrecht, Christine, Florian Kunneman, and Antal van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA'13)*, pages 29–37, Atlanta, GA.

Lin, Wei-Hao, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. 2006. Which side are you on?: Identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 109–116, Stroudsburg, PA.

Liu, Bing. 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Liu, Bing, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*, pages 342–351, Chiba.

Maynard, Diana and Mark Greenwood. 2014. Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, NV.

Miller, George A. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Mohammad, Saif M., Parinaz Sobhani, and Svetlana Kiritchenko. 2016. Stance and sentiment in tweets. *CoRR*, abs/1605.01655.

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

Nielsen, Finn Årup. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, volume 718, pages 93–98, Heraklion.

Poria, Soujanya, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, pages 1601–1612, Osaka.

Poria, Soujanya, Erik Cambria, Grégoire Winterstein, and Guang-Bin Huang. 2014. Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, 69:45–63.

Rajagopal, Dheeraj, Erik Cambria, Daniel Olsher, and Kenneth Kwok. 2013. A graph-based approach to commonsense concept extraction and semantic similarity detection. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*, pages 565–570, Rio de Janeiro.

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

Reyes, Antonio, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation*, 47(1):239–268.

Riloff, Ellen, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 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)*, pages 704–714, Seattle, WA.

Ritter, Alan, Sam Clark, Mausam and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of Empirical Methods for Natural Language Processing EMNLP*, pages 1524–1534.

Rosenthal, Sara, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin.

Scherer, Klaus R. 1999. *Handbook of Cognition and Emotion*, chapter: Appraisal Theory. John Wiley & Sons, Ltd.

Speer, Robert and Catherine Havasi. 2013. ConceptNet 5: A large semantic network for relational knowledge. In Iryna Gurevych and Jungi Kim, editors, *The People's Web Meets NLP: Collaboratively Constructed Language Resources*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 161–176.

Stone, Philip J., Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

Stranisci, Marco, Cristina Bosco, Delia Irazú Hernández Farías, and Viviana Patti. 2016. Annotating sentiment and irony in the online Italian political debate on #labuonascuola. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris.

Sulis, Emilio, Delia Irazú Hernández Farías, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. 2016. Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not. *Knowledge-Based Systems*, 108:132–143.

Toprak, Cigdem, Niklas Jakob, and Iryna Gurevych. 2010. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 575–584, Stroudsburg, PA.

Tsur, Oren, Dmitry Davidov, and Ari Rappoport. 2010. ICWSM—A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*, pages 162–169, Washington, DC.

Van de Kauter, Marjan, Diane Breesch, and Veronique Hoste. 2015. Fine-grained analysis of explicit and implicit sentiment in financial news articles. *Expert Systems with Applications*, 42(11):4999–5010.

Van de Kauter, Marjan, Geert Coorman, Els Lefever, Bart Desmet, Lieve Macken, and Véronique Hoste. 2013. LeTs Preprocess: The multilingual LT3 linguistic preprocessing toolkit. *Computational Linguistics in the Netherlands Journal*, 3:103–120.

Van de Kauter, Marjan, Bart Desmet, and Véronique Hoste. 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, Cynthia. 2017. *Can Machines Sense Irony? Exploring Automatic Irony Detection on Social Media*. Ph.D. thesis, Ghent University.

Van Hee, Cynthia, Marjan Van de Kauter, Orphée De Clercq, Els Lefever, and Véeronique Hoste. 2014. LT3: Sentiment classification in user-generated content using a rich feature set. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval'14)*, pages 406–410, Dublin.

Van Hee, Cynthia, Marjan Van de Kauter, Orphée De Clercq, Els Lefever, and Véronique Hoste. 2017. Noise or music? Investigating the usefulness of normalisation for robust sentiment analysis on social media data. *Traitement Automatique des Langues*, 58(1):63–87.

Van Hee, Cynthia, Els Lefever, and Véronique Hoste. 2016a. Guidelines for annotating irony in social media text, version 2.0. Technical report, LT3, Language and Translation Technology Team–Ghent University.

Van Hee, Cynthia, Els Lefever, and Véronique Hoste. 2016b. Monday mornings are my fave : #not. Exploring the automatic recognition of irony in English tweets. In *Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, pages 2730–2739, Osaka.

Van Hee, Cynthia, Els Lefever, and Véronique Hoste. 2016c. Exploring the realization of irony in Twitter data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1795–1799, Portorož.

Van Hee, Cynthia, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Véronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In *Proceedings of Recent Advances in Natural Language Processing*, pages 672–680, Hissar.

Veale, Tony and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 765–770, Amsterdam.

Wallace, Byron C. 2015. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4): 467–483.

Wang, Zelin, Zhijian Wu, Ruimin Wang, and Yafeng Ren. 2015. Twitter sarcasm detection exploiting a context-based model. In *Proceedings of the 16th International Conference on Web Information Systems Engineering (WISE'15)*, pages 77–91, Miami.

Wilson, Theresa. 2008. Annotating subjective content in meetings. In *Proceedings of the Sixth International Conference on Language*

Resources and Evaluation (LREC'08), pages 2738–2745, Marrakech.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 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, pages 347–354, Stroudsburg, PA.

Zhang, Lei and Bing Liu. 2011. Extracting resource terms for sentiment analysis. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP'11)*, pages 1171–1179, Chiang Mai.

Zhang, Meishan, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, 26th International Conference on Computational Linguistics*, pages 2249–2640, Osaka.