Harnessing Context Incongruity for Sarcasm Detection

Aditya Joshi[1,2,3] **Vinita Sharma**[1] **Pushpak Bhattacharyya**[1]
[1]IIT Bombay, India, [2]Monash University, Australia
[3]IITB-Monash Research Academy, India
aadi.cse@iitb.ac.in, pb@cse.iitb.ac.in

Abstract

The relationship between context incongruity and sarcasm has been studied in linguistics. We present a computational system that harnesses context incongruity as a basis for sarcasm detection. Our statistical sarcasm classifiers incorporate two kinds of incongruity features: explicit and implicit. We show the benefit of our incongruity features for two text forms - tweets and discussion forum posts. Our system also outperforms two past works (with F-score improvement of 10-20%). We also show how our features can capture inter-sentential incongruity.

1 Introduction

Sarcasm is defined as '*a cutting, often ironic remark intended to express contempt or ridicule*'[1]. Sarcasm detection is the task of predicting a text as sarcastic or non-sarcastic. The past work in sarcasm detection involves rule-based and statistical approaches using: (a) unigrams and pragmatic features (such as emoticons, etc.) (Gonzalez-Ibanez et al., 2011; Carvalho et al., 2009; Barbieri et al., 2014), (b) extraction of common patterns, such as hashtag-based sentiment (Maynard and Greenwood, 2014; Liebrecht et al., 2013), a positive verb being followed by a negative situation (Riloff et al., 2013), or discriminative n-grams (Tsur et al., 2010a; Davidov et al., 2010).

Thus, the past work detects sarcasm with specific indicators. However, we believe that it is time that sarcasm detection is based on well-studied linguistic theories. In this paper, we use one such linguistic theory: **context incongruity**. Although the past work exploits incongruity, it does so piecemeal; we take a more well-rounded view of incongruity and place it center-stage for our work. The features of our sarcasm detection system are based on two kinds of incongruity: '**explicit**' and '**implicit**'. The contribution of this paper is:

- We present a sarcasm detection system that is grounded on a linguistic theory, the theory of context incongruity in our case. Sarcasm detection research can push the frontiers by taking help of well-studied linguistic theories.
- Our sarcasm detection system outperforms two state-of-art sarcasm detection systems (Riloff et al., 2013; Maynard and Greenwood, 2014). Our system shows an improvement for short 'tweets' as well as long 'discussion forum posts'.
- We introduce inter-sentential incongruity for sarcasm detection, that expands context of a discussion forum post by including the previous post (also known as the 'elicitor' post) in the discussion thread.

Rest of the paper is organized as follows. We first discuss related work in Section 2. We introduce context incongruity in Section 3. Feature design for explicit incongruity is presented in Section 3.1, and that for implicit incongruity is in Section 3.2. We then describe the architecture of our sarcasm detection system in Section 4 and our experimental setup in Section 5. Quantitative evaluation is in Section 6. Inter-sentential sarcasm detection is in Section 7. Section 8 presents the error analysis. Section 9 concludes the paper and points to future directions.

2 Related Work

Sarcasm/irony as a linguistic phenomenon has been extensively studied. According to Wilson (2006), sarcasm arises from situational disparity. The relationship between context incongruity and sarcasm processing (by humans) has been studied in Ivanko and Pexman (2003). Several properties of sarcasm have also been investigated. Campbell

[1]Source: The Free Dictionary

and Katz (2012) state that sarcasm occurs along different dimensions, namely, failed expectation, pragmatic insincerity, negative tension, presence of a victim and along stylistic components such as emotion words. Eisterhold et al. (2006) observe that sarcasm can be identified based on the statement preceding and following the sarcastic statement. This is particularly true in cases where the incongruity is not expressed within the sarcastic text itself.

Computational detection of sarcasm is a relatively recent area of research. Initial work on sarcasm detection investigates the role of lexical and pragmatic features. Tepperman et al. (2006) present sarcasm recognition in speech using prosodic, spectral (average pitch, pitch slope, *etc.*) and contextual cues (laughter or response to questions). Carvalho et al. (2009) use simple linguistic features like interjection, changed names, *etc.* for irony detection. Davidov et al. (2010) train a sarcasm classifier with syntactic and pattern-based features. Gonzalez-Ibanez et al. (2011) study the role of unigrams and emoticons in sarcasm detection. Liebrecht et al. (2013) use a dataset of Dutch tweets that contain sarcasm-related hashtags and implement a classifier to predict sarcasm. A recent work by **?**) takes the output of sarcasm detection as an input to sentiment classification. They present a rule-based system that uses the pattern: if the sentiment of a tokenized hashtag does not agree with sentiment in rest of the tweet, the tweet is sarcastic, in addition to other rules.

Our approach is architecturally *similar* to Tsur et al. (2010b) who use a semi-supervised pattern acquisition followed by classification. Our feature engineering is based on Riloff et al. (2013) and Ramteke et al. (2013). Riloff et al. (2013) state that *sarcasm is a contrast between positive sentiment word and a negative situation*. They implement a rule-based system that uses phrases of positive verb phrases and negative situations extracted from a corpus of sarcastic tweets. Ramteke et al. (2013) present a novel approach to detect thwarting: the phenomenon where sentiment in major portions of text is reversed by sentiment in smaller, conclusive portions.

3 Context Incongruity

Incongruity is defined as '*the state of being not in agreement, as with principles*'[1]. Context incongruity is a necessary condition for sarcasm (Campbell and Katz, 2012). Ivanko and Pexman (2003) state that the sarcasm processing time (time taken by humans to understand sarcasm) depends on the degree of context incongruity between the statement and the context.

Deriving from this idea, we consider two cases of incongruity in sarcasm that are analogous to two degrees of incongruity. We call them **explicit incongruity** and **implicit incongruity**, where implicit incongruity demands a higher processing time. It must be noted that our system only handles incongruity between the text and common world knowledge (i.e., the knowledge that '*being stranded*' is an undesirable situation, and hence, '*Being stranded in traffic is the best way to start my week*' is a sarcastic statement). This leaves out an example like '*Wow! You are so punctual*' which may be sarcastic depending on situational context.

3.1 Explicit incongruity

Explicit incongruity is overtly expressed through sentiment words of both polarities (as in the case of '*I love being ignored*' where there is a positive word '*love*' and a negative word '*ignored*'). The converse is not true as in the case of '*The movie starts slow but the climax is great*'.

3.2 Implicit Incongruity

An implicit incongruity is covertly expressed through phrases of implied sentiment, as opposed to opposing polar words. Consider the example "*I love this paper so much that I made a doggy bag out of it*". There is no explicit incongruity here: the only polar word is '*love*'. However, the clause '*I made a doggy bag out of it*' has an implied sentiment that is incongruous with the polar word '*love*'.

3.3 Estimating prevalence

We conduct a naïve, automatic evaluation on a dataset of 18,141 sarcastic tweets. As a crude estimate, we consider an explicit incongruity as presence of positive and negative words. Around 11% sarcastic tweets have at least one explicit incongruity. We also manually evaluate 50 sarcastic tweets and observe that 10 have explicit incongruity, while others have implicit incongruity.

4 Architecture

Our system for sarcasm detection augments the feature vector of a tweet with features based on the

two types of incongruity. Specifically, we use four kinds of features: (a) **Lexical**, (b) **Pragmatic**, (c) **Implicit congruity**, and (d) **Explicit incongruity** features. Lexical features are unigrams obtained using feature selection techniques such as χ^2 Test and Categorical Proportional Difference. Pragmatic features include emoticons, laughter expressions, punctuation marks and capital words as given by Carvalho et al. (2009). In addition to the two, our system incorporates two kinds of incongruity features, as discussed next. The explicit incongruity features are numeric, qualitative features, while implicit incongruity features are related to implicit phrases.

4.1 Feature Design: Explicit Incongruity

An explicit incongruity giving rise to sarcasm bears resemblance to thwarted expectations (another commonly known challenge to sentiment analysis). Consider this example: '*I love the color. The features are interesting. But a bad battery life ruins it*'. The positive expectation in the first two sentences is thwarted by the last sentence. A similar incongruity is observed in the sarcastic '*My tooth hurts! Yay!*'. The negative word '*hurts*' is incongruous with the positive '*Yay!*'. Hence, our explicit incongruity features are a relevant subset of features from a past system to detect thwarting by Ramteke et al. (2013). These features are:

- Number of sentiment incongruities: The number of times a positive word is followed by a negative word, and vice versa
- Largest positive/negative subsequence: The length of the longest series of contiguous positive/negative words
- Number of positive and negative words
- Lexical Polarity: The polarity based purely on the basis of lexical features, as determined by Lingpipe SA system (Alias-i, 2008). Note that the '*native polarity*' need not be correct. However, a tweet that is strongly positive on the surface is more likely to be sarcastic than a tweet that seems to be negative. This is because sarcasm, by definition, tends to be caustic/hurtful. This also helps against humble bragging. (as in case of the tweet '*so i have to be up at 5am to autograph 7,000 pics of myself? Sounds like just about the worst Wednesday morning I could ever imagine*').

4.2 Feature Design: Implicit Incongruity

We use phrases with implicit sentiment as the implicit incongruity features. These phrases are sentiment-bearing verb and noun phrases, the latter being situations with implied sentiment (e.g. '*getting late for work*'). For this, we modify the algorithm given in Riloff et al. (2013) in two ways: (a) they extract only positive verbs and negative noun situation phrases. We generalize it to both polarities, (b) they remove subsumed phrases (i.e. '*being ignored*' subsumes '*being ignored by a friend*') while we retain both phrases. The benefit of (a) and (b) above was experimentally validated, but is not included in this paper due to limited space.

While they use rule-based algorithms that employ these extracted phrases to detect sarcasm, we include them as implicit incongruity features, in addition to other features. It is possible that the set of extracted situation phrases may contain some phrases without implicit sentiment. We hope that the limited size of the tweet guards against such false positives being too many in number. We add phrases in the two sets as count-based implicit incongruity features.

5 Experimental Setup

We use three datasets to evaluate our system:

1. **Tweet-A (5208 tweets, 4170 sarcastic)**: We download tweets with hashtags *#sarcasm* and *#sarcastic* as sarcastic tweets and *#notsarcasm* and *#notsarcastic* as non-sarcastic, using the Twitter API (`https://dev.twitter.com/`). A similar hashtag-based approach to create a sarcasm-annotated dataset was employed in Gonzalez-Ibanez et al. (2011). As an additional quality check, a rough glance through the tweets is done, and the ones found to be wrong are removed. The hashtags mentioned above are removed from the text so that they act as labels but not as features.
2. **Tweet-B (2278 tweets, 506 sarcastic)**: This dataset was manually labeled for Riloff et al. (2013). Some tweets were unavailable, due to deletion or privacy settings.
3. **Discussion-A (1502 discussion forum posts, 752 sarcastic)**: This dataset is created from the Internet Argument Corpus (Walker et al., 2012) that contains manual annota-

Lexical	
Unigrams	Unigrams in the training corpus
Pragmatic	
Capitalization	Numeric feature indicating presence of capital letters
Emoticons & laughter expressions	Numeric feature indicating presence of emoticons and 'lol's
Punctuation marks	Numeric feature indicating presence of punctuation marks
Implicit Incongruity	
Implicit Sentiment Phrases	Boolean feature indicating phrases extracted from the implicit phrase extraction step
Explicit Incongruity	
#Explicit incongruity	Number of times a word is followed by a word of opposite polarity
Largest positive /negative subsequence	Length of largest series of words with polarity unchanged
#Positive words	Number of positive words
#Negative words	Number of negative words
Lexical Polarity	Polarity of a tweet based on words present

Table 1: Features of our sarcasm detection system

tions for sarcasm. We randomly select 752 sarcastic and 752 non-sarcastic discussion forum posts.

To extract the implicit incongruity features, we run the iterative algorithm described in Section 4.2, on a dataset of 4000 tweets (50% sarcastic) (also created using hashtag-based supervision). The algorithm results in a total of 79 verb phrases and 202 noun phrases. We train our classifiers for different feature combinations, using LibSVM with RBF kernel (Chang and Lin, 2011), and report average 5-fold cross-validation values.

Features	**P**	**R**	**F**
Original Algorithm by Riloff et al. (2013)			
Ordered	0.774	0.098	0.173
Unordered	0.799	0.337	0.474
Our system			
Lexical **(Baseline)**	0.820	0.867	0.842
Lexical+Implicit	0.822	0.887	0.853
Lexical+Explicit	0.807	0.985	0.8871
All features	0.814	0.976	**0.8876**

Table 2: Comparative results for Tweet-A using rule-based algorithm and statistical classifiers using our feature combinations

6 Evaluation

Table 2 shows the performance of our classifiers in terms of Precision (P), Recall (R) and F-score (F), for Tweet-A. The table first reports values from a re-implementation of Riloff et al. (2013)'s two rule-based algorithms: the ordered version predicts a tweet as sarcastic if it has a positive verb phrase followed by a negative situation/noun phrase, while the unordered does so if the two are present in any order. We see that all statistical classifiers surpass the rule-based algorithms. The best F-score obtained is 0.8876 when all four kinds of features are used. This is an **improvement of about 5%** over the baseline, and 40% over the algorithm by Riloff et al. (2013). Table 3 shows that even in the case of the Discussion-A dataset, our features result in an improved performance. The F-score increases from 0.568 to 0.640, **an improvement of about 8%** in case of discussion forum posts, when all features are used.

Features	**P**	**R**	**F**
Lexical **(Baseline)**	0.645	0.508	0.568
Lexical+Explicit	0.698	0.391	0.488
Lexical+Implicit	0.513	0.762	0.581
All features	0.489	0.924	**0.640**

Table 3: Comparative results for Discussion-A using our feature combinations

To confirm that we indeed do better, we compare our system, with their reported values. This is necessary for several reasons. For example, we reimplement their algorithm but do not have

Approach	P	R	F
Riloff et al. (2013) (**best reported**)	0.62	0.44	0.51
Maynard and Greenwood (2014)	0.46	0.38	0.41
Our system (all features)	**0.77**	**0.51**	**0.61**

Table 4: Comparison of our system with two past works, for Tweet-B

access to their exact extracted phrases. Table 4 shows that we achieve a 10% higher F-score than the best reported F-score of Riloff et al. (2013). This value is also 20% higher than our re-implementation of Maynard and Greenwood (2014) that uses their hashtag retokenizer and rule-based algorithm.

7 Incorporating inter-sentential incongruity

Our system performs worse for Discussion-A than Tweet-A/B possibly because of incongruity outside the text. Because of the thread structure of discussion forums, sarcasm in a 'target post' can be identified using the post preceding it (called '*elicitor post*'), similar to human conversation (Eisterhold et al., 2006). For example, '*Wow, you are smart!*' may or may not be sarcastic. If a sarcasm classifier incorporates information from the elicitor post '*I could not finish my assignment*', a correct prediction is possible. Hence, we now explore how our incongruity-based features can help to capture '**inter-sentential incongruity**'. We compute the five explicit incongruity features for a concatenated version of target post and elicitor post (elicitor posts are available for IAC corpus, the source of Discussion-A). The precision rises to **0.705** but the recall falls to 0.274. A possible reason is that only 15% posts have elicitor posts, making the inter-sentential features sparse.

That notwithstanding, our observation shows that **using the inter-sentential context** is an interesting direction for sarcasm detection.

8 Error Analysis

Some common errors made by our system are:

1. **Subjective polarity**: The tweet '*Yay for 3 hour Chem labs*' is tagged by the author as sarcastic, which may not be common perception.
2. **No incongruity within text**: As stated in Section 2, our system does not detect sarcasm where incongruity is expressed outside the text. About 10% misclassified examples that we analyzed, contained such an incongruity.
3. **Incongruity due to numbers**: Our system could not detect incongruity arising due to numbers as in '*Going in to work for 2 hours was totally worth the 35 minute drive.*'.
4. **Dataset granularity**: Some discussion forum posts are marked as sarcastic, but contain non-sarcastic portions, leading to irrelevant features. For example, '*How special, now all you have to do is prove that a glob of cells has rights. I happen to believe that a person's life and the right to life begins at conception*'.
5. **Politeness**: In some cases, implicit incongruity was less evident because of politeness, as in, '*Post all your inside jokes on facebook, I really want to hear about them*'.

9 Conclusion & Future Work

Our paper uses the linguistic relationship between context incongruity and sarcasm as a basis for sarcasm detection. Our sarcasm classifier uses four kinds of features: lexical, pragmatic, explicit incongruity, and implicit incongruity features. We evaluate our system on two text forms: tweets and discussion forum posts. We observe an improvement of 40% over a reported rule-based algorithm, and 5% over the statistical classifier baseline that uses unigrams, in case of tweets. The corresponding improvement in case of discussion forum posts is 8%. Our system also outperforms two past works (Riloff et al., 2013; Maynard and Greenwood, 2014) with 10-20% improvement in F-score. Finally, to improve the performance for discussion forum posts, we introduce a novel approach to use elicitor posts for sarcasm detection. We observe an improvement of 21.6% in precision, when our incongruity features are used to capture inter-sentential incongruity.

Our error analysis points to potential future work such as: (a) role of numbers for sarcasm, and (b) situations with subjective sentiment. We are currently exploring a more robust incorporation of inter-sentential incongruity for sarcasm detection.

References

Alias-i. 2008. Lingpipe natural language toolkit.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in twitter, a novel approach. *ACL 2014*, page 50.

John D Campbell and Albert N Katz. 2012. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes*, 49(6):459–480.

Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio de Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

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

Dmitry Davidov, 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*, pages 107–116. Association for Computational Linguistics.

Jodi Eisterhold, Salvatore Attardo, and Diana Boxer. 2006. Reactions to irony in discourse: Evidence for the least disruption principle. *Journal of Pragmatics*, 38(8):1239–1256.

Roberto Gonzalez-Ibanez, 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: short papers-Volume 2*, pages 581–586. Association for Computational Linguistics.

Stacey L Ivanko and Penny M Pexman. 2003. Context incongruity and irony processing. *Discourse Processes*, 35(3):241–279.

CC Liebrecht, FA Kunneman, and APJ van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not.

Diana Maynard and Mark A Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of LREC*.

Ankit Ramteke, Pushpak Bhattacharyya, Akshat Malu, and J Saketha Nath. 2013. Detecting turnarounds in sentiment analysis: Thwarting. In *Proceedings of ACL*.

Ellen Riloff, 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 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714. Association for Computational Linguistics.

Joseph Tepperman, David R Traum, and Shrikanth Narayanan. 2006. yeah right : sarcasm recognition for spoken dialogue systems. In *INTERSPEECH*.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010a. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010b. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*.

Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *LREC*, pages 812–817.

Deirdre Wilson. 2006. The pragmatics of verbal irony: Echo or pretence? *Lingua*, 116(10):1722–1743.