Your Sentiment Precedes You: Using an author's historical tweets to predict sarcasm

Anupam Khattri[1] **Aditya Joshi**[2,3,4]
Pushpak Bhattacharyya[2] **Mark James Carman**[3]
[1]IIT Kharagpur, India, [2]IIT Bombay, India, [3]Monash University, Australia
[4]IITB-Monash Research Academy, India
anupam.khattri@iitkgp.ac.in ,{adityaj, pb}@cse.iitb.ac.in
mark.carman@monash.edu

Abstract

Sarcasm understanding may require information beyond the text itself, as in the case of 'I absolutely love this restaurant!' which may be sarcastic, depending on the contextual situation. We present the first quantitative evidence to show that historical tweets by an author can provide additional context for sarcasm detection. Our sarcasm detection approach uses two components: a contrast-based predictor (that identifies if there is a sentiment contrast within a target tweet), and a historical tweet-based predictor (that identifies if the sentiment expressed towards an entity in the target tweet agrees with sentiment expressed by the author towards that entity in the past).

1 Introduction

Sarcasm[1] is defined as '*the use of remarks that clearly mean the opposite of what they say, made in order to hurt someone's feelings or to criticize something in a humorous way*'[2]. An example of sarcasm is '*Being stranded in traffic is the best way to start my week*'(Joshi et al., 2015). There exists a sentiment contrast between the phrases '*being stranded*' and '*best way*' which enables an automatic sarcasm detection approach to identify the sarcasm in this sentence.

Existing approaches rely on viewing sarcasm as a contrast in sentiment (Riloff et al., 2013; Maynard and Greenwood, 2014). However, consider the sentences '*Nicki Minaj, don't I hate her!*' or '*I love spending four hours cooking on a weekend!*'. The sarcasm is ambiguous because of a likely hyperbole in the first sentence, and because sentiment associated with 'four hours cooking' depends on how much the author/speaker likes cooking. Such sarcasm is difficult to judge for humans as well as an automatic sarcasm detection approach. Essentially, we need more context related to the author of these sentences to identify sarcasm within them.

The question we aim to answer in this paper is: '**What sentiment did the author express in the past about the entities in the tweet that is to be classified? Can this information help us understand if the author is being sarcastic?**' We present the first quantitative evidence to show that historical text generated by an author may be useful to detect sarcasm in text written by the author. In this paper, we exploit the timeline structure of twitter for sarcasm detection of tweets. To gain additional context, we explore beyond the tweet to be classified (called '**target tweet**'), and look up the twitter timeline of the author of the target tweet (we refer to these tweets as the '**historical tweets**'). Our method directly applies to discussion forums and review websites, where other posts or reviews by this author may be looked at.

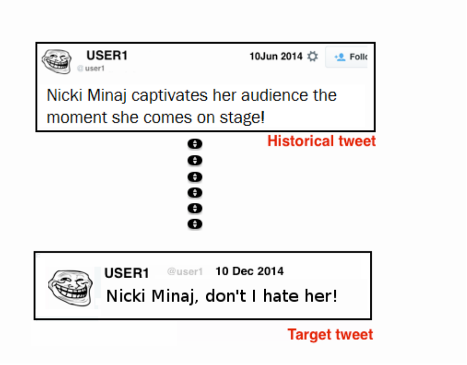
The rest of the paper is organized as follows. Section 2 contains the related work. We present a motivating example in Section 3, and describe the architecture of our approach in Section 4. The experimental setup and results are in Sections 5 and 6. We present a discussion of challenges observed with the proposed historical tweet-based approach in Section 7, and conclude the paper in Section 8.

2 Related work

Sarcasm detection relies mostly on rule-based algorithms. For example, Maynard and Greenwood (2014) predict a tweet as sarcastic if the sentiment embedded in a hashtag is opposite to sentiment in the remaining text. Similarly, Riloff et al. (2013) predict a tweet as sarcastic if there is a sentiment contrast between a verb and a noun phrase.

[1] We use irony and sarcasm interchangeably in this paper, as has been done in past work. Sarcasm has an element of criticism, while irony may not.

[2] http://dictionary.cambridge.org/dictionary/british/sarcasm


USER1 @user1 10Jun 2014 Foll

Nicki Minaj captivates her audience the moment she comes on stage!

Historical tweet

USER1 @user1 10 Dec 2014

Nicki Minaj, don't I hate her!

Target tweet

Figure 1: A motivating example for our approach

Similarly, supervised approaches implement sarcasm as a classification task that predicts whether a piece of text is sarcastic or not (Gonzalez-Ibanez et al., 2011; Barbieri et al., 2014; Carvalho et al., 2009). The features used include unigrams, emoticons, etc. Recent work in sarcasm detection deals with a more systematic feature design. Joshi et al. (2015) use a linguistic theory called context incongruity as a basis of feature design, and describe two kinds of features: implicit and explicit incongruity features. Wallace et al. (2015) uses as features beyond the target text as features. These include features from the comments and description of forum theme. In this way, sarcasm detection using ML-based classifiers has proceeded in the direction of improving the feature design, while rule-based sarcasm detection uses rules generated from heuristics.

Our paper presents a novel approach to sarcasm detection: '*looking at historical tweets for sarcasm detection of a target tweet*'. It is similar to Wallace et al. (2015) in that it considers text apart from the target text. However, while they look at comments within a thread and properties of a discussion thread, we look at the historical tweets by the author.

3 Motivating example

Existing approaches detect contrast in sentiment to predict sarcasm. Our approach extends the past work by considering sentiment contrasts beyond the target tweet. Specifically, we look at tweets generated by the same author in the past (we refer to this as '**historical tweets**'). Consider the example in Figure 1. The author USER1 wrote the tweet '*Nicki Minaj, don't I hate her?!*'. The author's historical tweets may tell us that he/she has spoken positively about Nicki Minaj in the past. In this case, we observe an additional tweet where the author describes having a good time at a Nicki Minaj concert. This additional knowledge helps to identify that although the target tweet contains the word '*hate*', it is sarcastic.

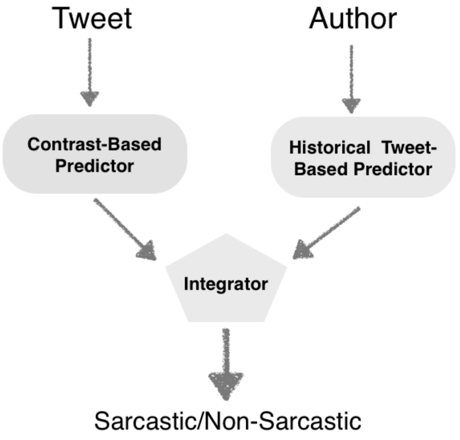


Figure 2: Architecture of our sarcasm detection approach

4 Architecture

Figure 2 shows the architecture of our sarcasm detection approach. It takes as input the text of a tweet and the author, and predicts the output as either sarcastic or non-sarcastic. This is a rule-based sarcasm detection approach that consists of three modules: (a) **Contrast-based Predictor**, (b) **Historical Tweet-based Predictor**, and (c) **Integrator**. We now describe the three modules in detail.

4.1 Contrast-based Predictor

This module uses only the target tweet. The contrast-based predictor identifies a sarcastic tweet using a sentiment contrast as given in Riloff et al. (2013). A contrast is said to occur if:

- **Explicit contrast**: The tweet contains one word of a polarity, and another word of another polarity. This is similar to explicit incongruity given by Joshi et al. (2015).
- **Implicit Contrast**: The tweet contains one word of a polarity, and a phrase of the other polarity. The implicit sentiment phrases are extracted from a set of sarcastic tweets as described in Tsur et al. (2010) Davidov et al. (2010). This is similar to implicit incongruity given by Joshi et al. (2015).

For example, the sentence '*I love being ignored.*' is predicted as sarcastic since it has a positive word '*love*' and a negative word '*ignored*'. We include rules to discount contrast across conjunctions like 'but' [3].

4.2 Historical Tweet-based Predictor

This module uses the target tweet and the name of the author. The goal of the historical tweet-based predictor is to identify if the sentiment expressed in the tweet does not match the historical tweets posted by the author. The steps followed are:

1. The sentiment of the target tweet is computed using a rule-based sentiment analysis system that we implemented. The system takes as input a sentence, and predicts whether it is positive or negative. It uses simple rules based on lookup in a sentiment word list, and rules based on negation, conjunctions (such as 'but'), etc. On Sentiment140 [4] corpus, our sentiment analysis system performs with an accuracy of 58.49%.
2. The target tweet is POS-tagged, and all NNP sequences are extracted as '**target phrases**'.
3. '**Target phrases**' are likely to be the targets of the sentiment expressed in the tweet. So, we *download only the historical tweets which contain the target phrases*[5].
4. The sentiment analysis system also gives the sentiment of the downloaded historical tweets. A majority voting-based sentiment in the historical tweets is considered to be the author's historical sentiment towards the target phrase.
5. This module predicts a tweet as sarcastic if the historical sentiment is different from the sentiment of the target tweet.

A target tweet may contain more than one target phrase. In this case, the predictor considers all target phrases, and predicts the tweet as sarcastic if the above steps hold true for any of the phrases. Possible lacunae in this approach are:

1. If the historical tweets contained sarcasm towards the target phrase, while the target tweet did not, the predictor will incorrectly mark the tweet as sarcastic.
2. If the historical tweets contained sarcasm towards the target phrase, and so did the target tweet, the predictor will incorrectly mark the tweet as non-sarcastic.
3. If an entity mentioned in the target tweet never appeared in the author's historical tweets, then no input from the historical tweet is considered.

4.3 Integrator

This module combines the predictions from the historical tweet-based predictor and the contrast-based predictor. There are four versions of the module:

1. **Only historical tweet-based**: This prediction uses only the output of the historical tweet-based predictor. This also means that if this author had not mentioned the target phrase in any of his/her tweets in the past, the tweet is predicted as non-sarcastic.
2. **OR**: If either of the two predictors marked a tweeet as sarcastic, then the tweet is predicted as sarcastic. If not, then it is predicted to be non-sarcastic.
3. **AND**: If both the predictors marked a tweet as sarcastic, then the tweet is predicted as sarcastic. If not, then it is predicted to be non-sarcastic.
4. **Relaxed-AND**: If both the predictors marked a tweet as sarcastic, then predict the tweet as sarcastic. If the historical tweet-based predictor did not have any tweets to look up (*i.e.,* the author had not expressed any sentiment towards the target in the past), then consider only the output of the contrast-based predictor.

5 Experimental Setup

For the contrast-based predictor, we obtain the implicit sentiment phrases as follows: (1) We download a set of 8000 tweets marked with #sarcasm, and assume that they are sarcastic tweets. These are not the same as the test tweets, (2) We extract 3-grams to 10-grams (1-gram represents a word) in these tweets, (3) We select phrases that occur at least thrice. This results in a set of 445 phrases. These phrases are used as implicit sentiment phrases for the contrast-based predictor.

For the historical tweet-based predictor, we first POS tag the sentence using Malecha and Smith

[3] For example, 'I like the movie but I dislike the cinema hall' does not count as a contrast, in terms of sarcastic expression

[4] http://help.sentiment140.com/for-students

[5] Twitter API allows access to the most recent 3500 tweets on a timeline. This is an additional limitation.

(2010). We then select NNP sequences[6] in the target tweet as the target phrase. Then, we download the complete timeline of the author using Twitter API [7], and select tweets containing the target phrase. The historical tweet-based predictor then gives its prediction as described in the previous section.

Both the predictors rely on sentiment lexicons: The contrast-based predictor needs sentiment-bearing words and phrases to detect contrast, while the historical tweet-based predictor needs sentiment-bearing words to identify sentiment of a tweet. We experiment with two lexicons:

1. Lexicon 1 (**L1**): In this case, we use the list of positive and negative words from Pang and Lee (2004).
2. Lexicon 2 (**L2**): In this case, we use the list of positive and negative words from Mohammad and Turney (2013).

Based on the two lexicons, we run two sets of experiments:

1. **Sarcasm detection with L1 (SD1)**: In this set, we use L1 as the lexicon for the two predictors. We show results for all four integrator versions (*Only historical tweet-based, AND, OR, Relaxed-AND*).
2. **Sarcasm detection with L2 (SD2)**: In this set, we use L2 as the lexicon for the two predictors. We show results for all four integrator versions (*Only historical tweet-based, AND, OR, Relaxed-AND*).

For all experiments, we use the test corpus given by Riloff et al. (2013). This is a manually annotated corpus consisting of 2278 tweets[8], out of which 506 are sarcastic.

6 Results

Tables 1 and 2 show Precision (P), Recall (R) and F-score (F) for SD1 and SD2 respectively. We compare our values with the best reported values in Riloff et al. (2013). This comparison is required because the test corpus that we used was obtained from them.

	P	**R**	**F**
Best reported value by Riloff et al. (2013)	0.62	0.44	0.51
Only Historical tweet-based	0.498	0.499	0.498
OR	0.791	**0.8315**	0.811
AND	0.756	0.521	0.617
Relaxed-AND	**0.8435**	0.81	**0.826**

Table 1: Averaged Precision, Recall and F-score of the SD1 approach for four configurations of the integrator

	P	**R**	**F**
Best reported value by Riloff et al. (2013)	0.62	0.44	0.51
Only Historical tweet-based	0.496	0.499	0.497
OR	0.842	**0.927**	**0.882**
AND	0.779	0.524	0.627
Relaxed-AND	**0.880**	0.884	**0.882**

Table 2: Averaged Precision, Recall and F-score of the SD2 approach for four configurations of the integrator

Table 1 shows that using only the historical tweet-based predictor, we are able to achieve a comparable performance (F-score of approximately 0.49 in case of SD1 and SD2 both) with the benchmark values (F-score of 0.51 in case of Riloff et al. (2013)). The performance values for 'Only historical tweet-based' are not the same in SD1 and SD2 because the lexicon used in predictors of the two approaches are different. This is obviously low because only using historical contrast is not sufficient.

The AND integrator is restrictive because it requires both the predictors to predict a tweet as sarcastic. In that case as well, we obtain F-scores of 0.617 and 0.627 for SD1 and SD2 respectively. Relaxed-AND performs the best in both the cases with F-scores of 0.826 and 0.882 for SD1 and SD2 respectively.

We experiment with two configurations SD1 and SD2, in order to show that the benefit of our approach is not dependent on the choice of lexicon. To understand how well the two captured the positive (*i.e.,* sarcastic tweets) class, we compare their precision and recall values in Table 3. We

[6] We also experimented with NN and JJ_NN sequences. However, the output turned out to be generic.

[7] `https://dev.twitter.com/overview/api`

[8] Some tweets in their original corpus could not be downloaded due to privacy settings or deletion.

observe that the positive precision is high in case of OR, AND, Relaxed-AND. The low precision-recall values in case of 'Only historical tweet-based' indicates that relying purely on historical tweets may not be a good idea. The positive precision in case of Relaxed-And is 0.777 for SD1 and 0.811 for SD2. The contrast within a tweet (captured by our contrast-based predictor) and the contrast with the history (captured by our historical tweet-based predictor) both need to be applied together.

7 Discussion

Our target phrases are only NNP sequences. However, by the virtue of the POS tagger[9] used, our approach predicts sarcasm correctly in following situations:

1. **Proper Nouns**: The tweet '*because **Fox** is well-balanced and objective?*' was correctly predicted as sarcastic because our predictor located a past tweet '***Fox**'s World Cup streaming options are terrible*'.

2. **User Mentions**: User mentions in a tweet were POS-tagged as NNPs, and hence, became target phrases. For example, a target tweet was '***@USERNAME** ooooh that helped alot*', where the target phrase was extracted as @USERNAME. Our approach looked at historical tweets by the author containing '*@USERNAME*'. Thus, the predictor took into consideration how 'cordial' the two users are, based on the sentiment in historical tweets between them.

3. **Informal Expressions**: Informal expressions like 'Yuss' were tagged as NNPs. Hence, we were able to discover the common sentiment that were used in, by the author. The target tweet containing 'Yuss' was correctly marked as sarcastic.

However, some limitations of our approach are:

1. **The non-sarcastic assumption**: We assume is that the author has not been sarcastic about a target phrase in the past (because we assume that the historical tweets contain an author's 'true' sentiment towards the target phrase).

2. **Timeline-related challenges**: Obtaining the Twitter timeline of an author may not be straightforward. A twitter timeline may be private where the user adds his/her followers, and only these followers have access to the user's tweets. Twitter also allows change of twitter handle name because of which the timeline cannot be searched. In some cases, the twitter account was deactivated. Hence, we could not download the twitter timeline for 248 out of 2273 unique authors in our dataset.

	SD1 PP	**SD1 PR**	**SD2 PP**	**SD2 PR**
OHTB	0.218	0.073	0.215	0.063
OR	0.647	0.785	0.691	0.978
AND	0.727	0.047	0.771	0.053
Relaxed-AND	0.777	0.675	0.811	0.822

Table 3: Positive Precision (PP) and Recall (PR) for SD1 and SD2; OHTB: Only Historical tweet-based

8 Conclusion & Future Work

Past work in sarcasm detection focuses on target tweet only. We present a approach that predicts sarcasm in a target tweet using the tweet author's historical tweets. Our historical tweet-based predictor checks if the sentiment towards a given target phrase in the target tweet agrees to the sentiment expressed in the historical tweets by the same author. We implement four kinds of integrators to combine the contrast-based predictor (which works on the target tweet alone) and the historical tweet-based predictor (which uses target tweet and historical tweets). We obtain the best F-score value of 0.882, in case of SD2, where the contrast predictor uses a set of polar words from a word-emotion lexicon and phrases with implicit sentiment.

Our work opens a new direction to sarcasm detection: considering text written by an author in the past to identify sarcasm in a piece of text. With availability of such data in discussion forums or social media, sarcasm detection approaches would benefit from making use of text other than just the target text. Integration of historical text-based features into a supervised sarcasm detection framework is a promising future work.

[9] For example, some POS taggers have a separate tag for user mentions.

References

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

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.

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.

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.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *ACL-IJCNLP 2015*.

Gregory Malecha and Ian Smith. 2010. Maximum entropy part-of-speech tagging in nltk. *unpublished course-related report*.

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

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

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

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.

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

Byron C Wallace, Do Kook Choe, and Eugene Charniak. 2015. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *ACL-IJCNLP 2015*.