ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

SARCASM DETECTION

Tulasi Prasad Sariki, G. Bharadwaja Kumar, Nishant Kambhatla
School of Computing Science and Engineering, Vellore Institute of Technology, Chennai - 600127, India.
Email: tulasiprasad.sariki@vit.ac.in

Received on: 15.10.2016 *Accepted on: 12.11.2016*

Abstract

Sarcasm is an indirect form of speech intentionally used to express a witty remark implicitly rather than its literal meaning. It is a phenomenon very common in social media. Because of its inherently ambiguous nature, detecting sarcasm is a difficult task not only for computers but also for humans. We present a novel computational approach that harnesses the situational and contextual disparity coupled with a sarcasm-indicating lexicon as a basis for sarcasm detection. Our investigations in various factors that indicate situational disparity culminated in Blunt Sentiment Contrast, Word Polarity Contrast, Word Phrase Polarity Contrast and Inter Phrase Polarity Contrast. Alongside, we compiled a lexicon of patterns cuing sarcasm from various sources and assigned weights by 4 different evaluators manually. We also improvised an antecedent bootstrapping algorithm which facilitates automatic learning for contrasting situational phrases. We logically combined the predictions of sarcastic instances given by the feature based SVM classifiers, and those of the pattern matching approach based weighted-lexicon.

We then conducted experiments on a collection of 137,700 tweets collected from Twitter and obtained an average F-score of 0.93. To validate our approach, we carried out tests on a gold standard test set used in Riloff et al.'s work. We obtained an improvement about 28% over the best results reported in the literature.

Keywords: computational linguistics, sarcasm, weighted-lexicon, situational disparity, contextual contrast, sarcasm detection.

1. Introduction

SARCASM is defined as the use of words that mean the opposite of what you really want to say especially in order to insult someone, to show irritation, or to be funny. Though evinced in many ways, identifying sarcasm is a difficult yet

important task in opinion mining. This is because the computational systems which rely on or exploit polarity of the text can be misled by sarcastic sentences where the literal sentiment of the text is different from the one that is implied, often involving a strand of hostility (Schwoebel, Dews, Winner and Srinivas, 2000)[16].

Consider the following tweet from Twitter: " I'm glad I have to work today so all people celebrating Labor Day can enjoy their holiday #sarcastic. "

The tweet contains positive polar words glad and celebrating which can lead a sentiment analyzer to determine that the tweet carries a positive sentiment.

However, it clearly has a negative sentiment. Luckily, in this instance, the hashtag #sarcasm divulges the implied sarcasm but that may not be the case always. This type of sophistication in the use of words where the message is conveyed implicitly makes sarcasm hard to recognize.

The difficulty in spotting sarcasm has a direct impact on NLP systems ranging from review summarization systems to chatbots owing to the lack of success of the state-of-the-art sentiment analyzers in properly interpreting sarcasm. Sarcasm detection has not received much attention even though non literal language, for instance metaphor, has become a much sought after topic in computational linguistics. However, sarcasm has been studied well in the context of psychology and cognitive sciences. Gerrig et al. (Gerrig and Goldvarg, 2000) describes that the perception of sarcasm increases with an increase in the size of situational disparity[7]. According to Ellestrom et al. (Ellestrom, 2002, pp.51) situational disparity is the key reason behind situational irony and is defined as a state where the outcome is incongruous with what was expected[6].

The conflation of sarcasm and irony (Nakassis and Snedeker, 2002, pp. 429–440) makes it easier to look at the both as assertions with positive literal meanings and negative intended meanings (Creusere, 1999[) 13, 4.] This allows us to employ the salient features of both irony and sarcasm interchangeably (Lee and Katz, 1998[) 10.]

Among the several works that have been carried out in sarcasm detection in computational linguistics, the most closely related to the work presented in this paper is that of Riloff et al. (Riloff, Qadir, Surve, De Silva, Gilbert and Huang, 2013) which shows that sarcasm can be spotted by the contrast between a positive sentiment word and a negative state[15]. SASI algorithm given by Tsur et al. (Tsur, Davidov and Rappoport, 2010) acquires semi supervised patterns that identify sarcastic occurrences[19].

This algorithm has catalyzed the work presented in this paper in compiling, refining and using a lexicon of sarcasm identifying words and phrases which combined with their assigned weights adds to the probability of a tweet being sarcastic. Consider the following set of tweets in **Table 1** from our corpus:

Table 1. Some tweets from our corpus

(1) EARLY MORNING DOCTOR APPOINTMENTS ARE JUST A BUNDLE OF FUN #SARCASM.
(2) THERE IS NOTHING BETTER THAN ACCIDENTALLY GETTING TOOTHPASTE ON YOUR SHIRT.#SARCASM
(3) I SO LOVE WHEN PEOPLE GIVE UNSOLICITED OPINIONS. #SARCASM
(4) YAY! I AM SICK #SARCASM
(5) WHAT A WAY TO RUIN MY NIGHT. WOW! THANK YOU SO MUCH...REALLY #SARCASM
(6) ITS HILARIOUS HOW LITTLE I?M TAKEN SERIOUSLY #SARCASM
(7) HOW COZY TO EXPERIENCE ALL THE EXCITEMENT OF THE RUSH HOUR TRAFFIC IN THE MALL PARKINGLOT. # SARCASM
(8) OH YES HOUSE ARREST IS SUCH A DELIGHT #SARCASM
(9) #KYLOREN IS THE BEST VILLAIN #STARWARS HAS EVER GIVEN US #SARCASM
(10) I WAS HAVING SO MUCH FUN DOING END-OF-TERM GRADE REVIEWS. #SARCASM

Examples (1), (2), (3), (7), (10) clearly have a disparity between the positive sentiment conveyed and the negative sentiment intended. In (10) “having so much fun” is the phrase that conveys centrally positive sentiment while doing end-of-term grade reviews is the phrase indicating the negative situation. This situational disparity or the contextual contrast is therefore a key factor in determining whether or not a tweet is sarcastic.

However, examples (1), (2), (3), (5), (6), (8), (9) exhibit a prime linguistic feature helpful in spotting sarcasm - hyperbole. Hyperbole is defined as the use of exaggeration as a rhetorical device for emphasis. The phrases “bundle of fun”, “nothing better than”, “so love”, “hilarious”, “such a delight”, “the best villain ... ever ” are purely an overstatement.

The unexpected usage of positive interjections like “Yay!” and “WOW!” in a negative situation as in examples (4) and (5) also show sarcasm. There are certain words, phrases and references which are used in, especially, the social media to show sarcasm. Consider the tweets in **Table 2** from our corpus along with the cue words/patterns/phrases.

Table 2. Word/Phrase Patterns used in tweets to show Sarcasm.

(A) Monday
(a) And Monday is already off to a fabulous start.. #nope #sarcasm

(b) You knew there had to be one of these days! #Monday #sarcasm

(c) Show the world you are "fine and have no problem"!! Start a new day!! **It's**Monday!! #sarcasm

(d) Its Monday. The "bad atmosphere" is about to return back to work. I am really excited about today #sarcasm

(B) Weekend ...Work ... Study

(a) What a fun filled Saturday...of work! Yay work on Sunday too..:(#sarcasm #work #weekend

(b) Waiting for the weekend because being stuck in a boring ass office is SO MUCH FUN.. #sarcasm

(c) It was nice to have a 3 day weekend but I think we are all ready for work tomorrow #sarcasm

(d) Annnnd guess I should start on my homework now #weekend #sarcasm

(C) Didn't see that coming

(a) But yeah, didn't take long for "fans"to moan that the remake isn't what they wanted. Didn't see that coming. #sarcasm

(b) Kobe retires after the season. Totally didn't see that coming. #sarcasm

(c) Halo 5 sold badly? Totally didn't see that coming. Nope. Not at all. #sarcasm

(d) Whoa another Walking Dead maze, didn't see that one coming #sarcasm

The tweets in Table 2 are examples of how some specific word patterns ("Monday", "weekend + work/study", "didn't see that coming") are used in the context of social media that are intended to be sarcastic more often than we can imagine. We compiled a list of such common cues which are almost extensively used to show sarcasm particularly in social media and used it as a secondary basis for classifying sarcastic tweets.

In this paper, we present a novel approach for detecting sarcasm in the realm of Twitter. Our algorithm consists of three key components: a) identifying features responsible for situational disparity, b) identifying lexical, syntactic and pragmatic features, and c) a lexicon of sarcasm identifying cue patterns. Several experiments were conducted to decide

on how to use the lexicon efficiently and this resulted in a weighted-lexicon with weights assigned by four different evaluators. We use positively sarcastic tweets for the learning purpose. Bootstrapping culminates in a list of positive sentiment phrases and negative state phrases, and vice-versa. We use this list of phrases to extract features responsible for situational disparity. We experiment to identify sarcasm on a gold standard test set of 2232 tweets. Our algorithm outperforms Riloffetals (Riloff et al., 2013) system[15].

2. Related Work

Sarcasm detection approaches in computational linguistics have mostly been rule-based. Maynard et al. (Maynard and Greenwood, 2014) exploits the contrast in the sentiment embedded in the hashtag to that of the remaining tweet to predict it as sarcastic [12]. In the context of spoken dialog systems Tepperman et al. (Tepperman, Traum and Narayanan, 2006) has relied primarily on speech-related cues like prosody and laughter [18]. The role of lexical features such as interjections (such as "Yay!") and punctuations (such as "!!") have also been investigated to identify sarcasm (Carvalho, Sarmento, Silva and de Oliveira, 2009)[2.] Bootstrapping method to learn the lexical cues and other lexico-syntactic features associated with sarcasm were also explored (Lukin and Walker, 2013). Riloff et al. (Riloff et al., 2013) classifies a tweet as sarcastic if the sentiment of the verb phrase is the opposite to that of the noun phrase [11, 15]. Tsur et al. (Tsur et al., 2010) proposed SASI, a semi supervised framework that harnesses the syntactic and pattern based features in sarcastic text [19].

Taboada et al. (Taboada, Brooke, Tofiloski, Voll and Stede, 2011) showed how a lexicon can be effectively used for performing sentiment analysis [17]. Gonzalez-Ibanez et al. (Gonzalez-Ibanez, Muresan and Wacholder, 2011) investigated the usefulness of lexical and pragmatic features in detecting sarcasm [9]. Davidov et al. (Davidov, Tsur and Rappoport, 2010) conducted experiments by using large number of tweets and Amazon reviews to determine the reliability of tweets with hashtag #sarcasm as a gold standard to test for sarcasm [5]. It was, however, found that such tweets are noisy.Our approach of employing situational disparity in the tweets is partly inspired by Riloff et al. (Riloff et al., 2013)[) 15.] We create the baseline system and optimize it to include more lexical and syntactic features.

3. Data

Twitter is a popular micro-blogging platform that allows people to publish short messages called tweets. The length of a tweet is restricted to 140 characters. Apart from text, a tweet can contain hashtags (#example) which is assigned by the

user to establish the topic, mood or sentiment of the tweet, user mentions (@username) and URLs. We used a Twitter API to mine 200,000 unique tweets consisting of both sarcastic and non-sarcastic tweets. Assuming the best judge of whether a sense of sarcasm is carried in a tweet is its author, we relied on the annotations assigned to the tweets in the form of hashtags. We used the hashtags #sarcasm and #sarcastic to collect 120,000 sarcastic tweets and the hashtags #happy, #sad and #quote to collect 80,000 non-sarcastic tweets of mixed sentiments. Building a corpus of tweets which are already annotated is helpful for two main reasons – it saves the tediousness of manually labeling such a huge dataset and it overcomes the issue of accuracy in manually labeling the dataset.

Data Preprocessing: In order to make our corpus more effective, we applied a filter to identify and prune tweets that had user-mentions and URLs. Tweets that were duplicates, retweets or in languages other than English were also filtered. A hashtag filter was applied to make sure that only tweets with hashtags at the end were allowed. Any tweet which had hashtag(s) as a part of the message rather than an annotation were removed. This eliminated messages such as "I didn't use #sarcasm in my last response but I should have". We then performed a manual review of the sarcastic dataset to remove any tweets that had the hashtag at the end, but were about sarcasm rather than being sarcastic. Thus, tweets like "I love how someone can respond to your sarcasm by #sarcasm." were eliminated.

4. Architecture

The architecture of our approach for sarcasm detection is shown in **Figure 1**. Our method requires preprocessed tweets as input and classifies each of them as sarcastic or non-sarcastic. Our system is composed of: (1) Feature Identification: a) Situational Disparity Features; b) Lexical, Syntactic and Pragmatic Features, (2) Sarcasm Prediction: a) Feature Based (SVM) classifier; b) Weighted-Lexicon based Sarcasm identifier and (3) Logical integration of the two predictors.

The following sub-sections contain the description of each module.

4.1. Feature Identification: (a) Situational Disparity

Disparity means disagreement or the quality or state of being different. Situational Disparity (SD) is a necessary condition for sarcasm (Campbell and Katz, 2012[) 1.] This disagreement between the central positive sentiment of the statement and the referred negative situation constitutes the first module of our computational system to automatically identify sarcasm. We implemented several different forms to identify relevant features to spot SD:

A. Blunt Sentiment Contrast: This is a naive and simple two-step process. First, we perform a sentiment analysis of the tweet as a whole. Second, ignoring the neutral tweets, we find tweets that are positive and contain word(s) of negative polarity, and vice-versa, and classify them as SD.

B. Word Polarity Contrast: This is a slightly better method than the one described above. Instead of analyzing the sentiment of the tweet, we break down the tweet into a set of words and perform sentiment analysis of each word. If a tweet contains positive words and negative words, it is classified as SD.

C. Word-Phrase Polarity Contrast : We extract tacit sentiment phrases from a tweet using a semi-supervised pattern matching method similar to the one described in Tsur et al. (Tsur et al., 2010) [19]. If a tweet contains a word of a polarity and a phrase of a different polarity, it is marked as SD.

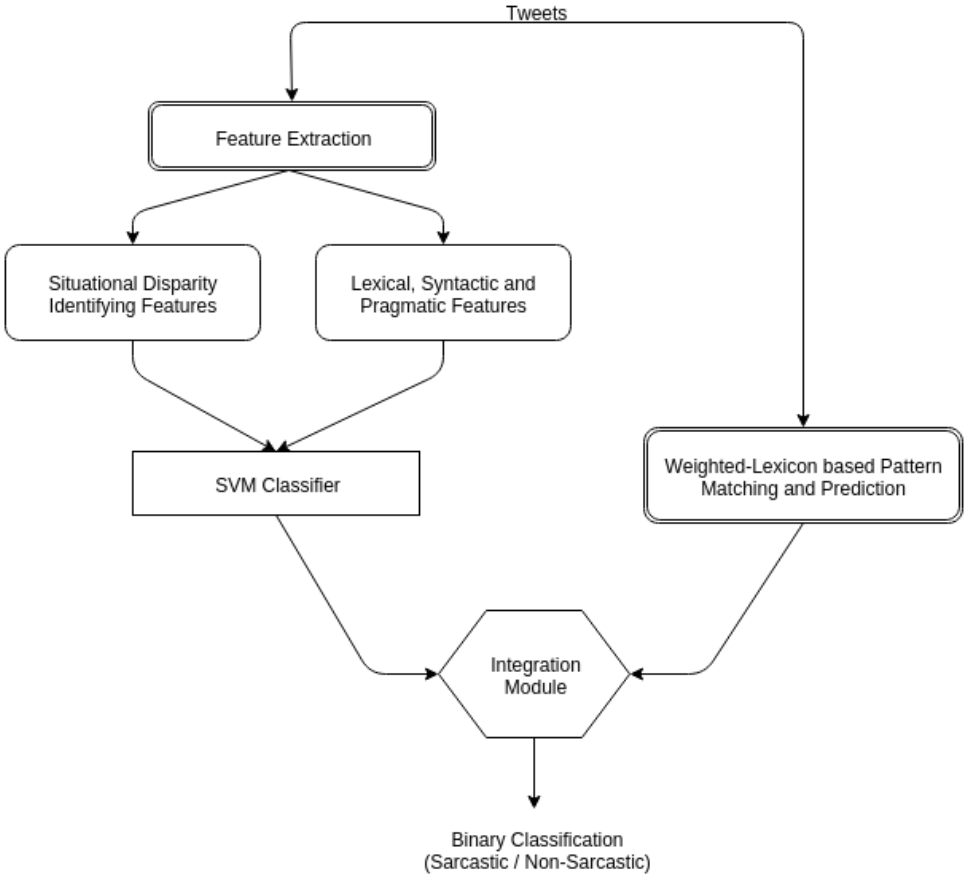

Figure 1. Architecture of our approach for sarcasm detection.

D. Phrase-Phrase Polarity Contrast: This form is of utmost interest to our approach. Tweets are split into sentiment-conveying verb phrases and noun phrases. The noun phrases are the ones that depict situations bearing a sentiment. The disagreement between the sentiment of a verb phrase and a noun phrase clearly defines a situational disparity. We improvise the bootstrapping method described in Riloff et al. (Riloff et al., 2013) in two ways: a) they only used positive

seed words to extract negative situational noun phrases [15]. We extend it to include both polarities - Positive verb phrase and Negative noun phrase pairs, and Negative verb phrases and Positive noun phrase pairs. b) Lemmatization: Since bootstrapping is primarily concerned with extracting verb phrases while building an exhaustive list of positive and negative word sets, it can be posited that lemmatization (converting verb tenses to their base forms) is suited to bootstrapping than stemming (which operates on all words).Rather than using the set of words and phrases extracted to detect sarcasm directly, we use them as features for identifying SD.

Leveraging these 4 factors for identifying Situational Disparity, we perform a feature construction detect sarcasm. We extract the cue words, word-word pairs, word-phrase pairs, and include them as SD Features in addition to the verb-noun phrase pairs extracted by bootstrapping.

Table 3. Features for sarcasm detection based on Situational Disparity.

Situational Disparity (SD) Features	
Feature	**Description**
Positive Words	Words with positive polarity
Negative Words	Words with negative polarity
Polar Word Pairs	Common word pairs with different polarity
Positive Phrases	Positive situation phrases
Negative Phrases	Negative situation phrases
Tweet Polarity	Overall sentiment of tweet
#Positive Words	No. of words with positive polarity
#Negative Words	No. of words with negative polarity
#Polarity Contrast	No. of times contrast in polarity is observed

4.2. Feature Identification: (b) Lexical, Syntactic and Pragmatic

In this subsection, we describe the lexical, syntactic and pragmatic features used in order to strengthen the feature-set that we used to recognize sarcasm (**Table 4**).

A.Lexical Features: Three types of lexical features were used -unigrams, bigrams and resource-based. All the lexical features were obtained using chi-square feature selection technique. For resource based lexical feature selection, in

particular, we used LIWC dictionary (Pennebaker, Chung, Ireland, Gonzales and Booth, 2007) which has a set of word categories grouped into Linguistic Processes, Psychological Processes, Personal Concerns and Spoken Categories [14]. Besides Chi-Square, we also used Categorical Proportional Difference upon the LIWC list to make the feature selection better.

B. Syntactic Features: The only syntactic feature we used is the average lengths of the tweets in the sarcastic and the non-sarcastic datasets.

C. Pragmatic Features: Several pragmatic features have been identified as cues to sarcasm. We used CMU's POS Tagger for Tweets (Gimpel, Schneider, O'Connor, Das, Mills, Eisenstein, Heilman, Yogatama, Flanigan and Smith, 2011), which supports internet shorthands as well as emoticons [8]. We include the following features: (1) Ellipsis...: Using it may change the tone and meaning of what is written. The tweet "No Jordan Reed isn't Pro Bowl Worthy... Not even close #sarcasm" shows how ellipsis is generally used, (2) Punctuation Marks: Especially multiple interjection marks are widely used in the context of social media and they are often indicative of sarcasm, (3) Laughter Slangs: Internet slangs specifically related to expression of laughter such as LOL, ROFL, LMAO etc, (4) Emoticons: Tweets are almost incomplete without emoticons. These emoticons add to the local and global sentiment of the tweet, (5) Interjections: (oh, aah, huh, yeah, awesome etc), (6) Capitalization: Examples like "Oh! Thats REALLY helpful #sarcasm" show how upper-case is used to emphasize sarcasm, and (7) Hyperbole: It is defined as an intentional extravagant exaggeration. An example from our corpus is "This is epic! Can't get to school #sarcasm". In the tweet this is epic is a hyperbole emphasizing sarcasm. We compiled a list of most commonly used hyperbole 2 and extracted patterns from the list and added them to our list of features to identify sarcasm. The token overlap of the words in our hyperbole list with the words in all the tweets was 27%.

Table 4. Lexical, Syntactic and Pragmatic Features for sarcasm detection.

Feature	Description
Lexical Features	
Unigrams (n=1) Bigrams (n=2)	N-grams from our training corpus
Syntactic Features	
Average lengths of the tweets	The average number of words in the

	Sarcastic tweets and Non-Sarcastic tweets.
Pragmatic Features	
Ellipsis Punctuation Marks Laughter Slangs Emoticons Interjections Capitalization Hyperbole	Boolean feature indicating presence of the each of the features (ellipsis, punctuation marks, laughter slangs, and so on) respectively

4.3. Sarcasm Prediction: (a) Feature Based

The features extracted in the above subsections 4.1 and 4.2 are used in different combinations in our support vector machine (SVM) classifier using LibSVM (Chang and Lin, 2011) to classify tweets as sarcastic and non-sarcastic [3]. More on this is discussed in section 5.

4.4. Sarcasm Prediction: (a) Weighted-Lexicon based

As a comprehensive approach towards detecting sarcastic utterances at a human level, we extensively studied 50,000 tweets that had the hashtag #sarcasm or #sarcastic to analyse any resemblance among them. Similar study was conducted upon 50,000 tweets that did not have #sarcasm / #sarcastic hashtag. The juxtaposition of the two studies facilitated our understanding that certain words and phrases were almost exhaustively used to cue sarcasm in tweets. For example, 373 out of 438 occurrences of the phrase "Yeah right!" were found to be in tweets that were sarcastic. We compiled all such words and phrases into a list. The weighted-lexicon has phrases with:

Confidence Score, c:

To validate our hypothesis that these phrases could be treated as cue phrases for sarcasm, we generated a confidence score (c) for each of the cue pattern in the list using Levenshtein Distance on each tweet from our corpus of sarcastic tweets. The confidence score (c) was a decimal generated between 0 and 1 achieved by normalizing the Levenshtein Distance value obtained, 1 implying the most confidence.

Cue weight, w:

The lexicon we compiled consists of 72 sarcasm cuing patterns. The lexicon obviously had some patterns that are quite frequently used in the context of sarcasm and some that are less often indicative of sarcasm. To address this issue, we

employed 4 human evaluators to allot weights between 0 and 5 to the expressions and patterns in the lexicon with possible increments of 0.5, 5 indicating highest relevance and affinity towards sarcasm. The inter evaluator agreement for 72 assignments was measured to a Fleiss' kappa $\kappa = 0.59$. Thus, each cue phrase in the lexicon had 4 weights assigned by 4 different evaluators. To make the weight more accessible, we assigned the final weight (w) as the median of the 4 weights allotted by the evaluators. **Table 5** below showsa small part of the final weighted-lexicon:

Table 5. A part of the Weighted-Lexicon for sarcasm detection.

Patterns/Expressions	**Weight (w)**	**Confidence (c)**
yeah right	5	0.93
how would you like your [holiday/vacation/weekend]+ [work/office/study/homewor k]	5	0.94
good luck with that	4.5	0.70
didn't see that coming	4	0.75
yay!	2	0.61
to that i say	3	0.4
what a great way to	2	0.55
.	.	.
.	.	.

Interestingly, as logically conclusive from the third row of the above Table 5, mention of holiday/weekend and work/study in the same tweet has consistently coincided with tweet being sarcastic.

Normalized Confidence, η: We normalize the weight (w) and confidence (c) by rescaling both into a unified scale of (0, 1). To achieve the normalized score, we simply change each of these into a scale of 0.5 and add them up. Therefore, the normalized confidence is given by:

$$\eta = w/10 + c//2 \qquad \text{(Equation 4.1)}$$

Wherew is the weight, and c is the confidence score.

Fulfillment Score, f: To find the extent to which the patterns in the lexicon match with cue patterns in the tweets, we calculate the fulfillment score (f) by a two-step process: (1) Firstly, a fuzzy pattern matching with each target phrase

from the lexicon over the tweet. This gives us, if found, the approximate pattern in the tweet that is the closest match which we refer to as a kin pattern. (2) Secondly, Fulfillment Score (f) is calculated as:

f =Jaro – Winkler Distance (cuepattern, kinpattern) (Equation 4.2)

Wherecuepattern ϵ lexicon, and kinpattern ϵ tweet as calculated in (1). The fulfillment score (f) is a decimal number between 0 and 1.

Sarcasm Prediction: To predict the sarcasm, we first extract the respective lengths of the cue pattern in the lexicon (P), kin pattern in the tweet (), the tweet itself () and the average length of the sarcastic tweets in our corpus (L). We then calculate the Utility Vaue (γ) as:

$$\gamma = \left[1 - \left|\frac{P}{L} - \frac{p}{l}\right|\right]\left[\frac{2f\eta}{f+\eta}\right]$$ (Equation 4.3)

Where P is the length of cue pattern in lexicon,

p is the length of kin pattern in the tweet,

L is the average length of tweets in our corpus,

l is the length of the tweet.

Equation (4.3) assumes that we find only one cue pattern in one tweet. Consider a sarcastic tweet from our corpus:

“yeah right, better having kids with a rifle in their hands killing their classmates, than loving their classmates.. #sarcasm”

The kin pattern matched in this tweet is yeah right. For this we measured the normalized confidence η = 0.965 as described in Equation (1) and the fulfillment score f = 0.92 as in Equation (2). The lengths of the cue pattern and the kin pattern are P = 2 and p = 2 respectively. The average length of the sarcastic tweets was measured to be L = 11 and the length of tweet = 18. Using Equation (3) the utility value γ = 0.875.

However unlikely, to address a situation where more than 1 pattern exists in a tweet, we slightly modify our calculation of γ. We use the following equation to calculate the overall Utility Value:

$$\gamma = \frac{\sum_{i=1}^{n} w_i \gamma_i}{\sum_{i=1}^{n} w_i}$$ (Equation 4.4)

Wheren is the number of unique patterns found,

w_i is the corresponding weight of a pattern,

γ_i is the corresponding utility value of a pattern.

We use a threshold for the Utility Value (γ) to classify a tweet as sarcastic or non-sarcastic. If the overall γ of a tweet is equal to or more than 0.5, it is classified as a sarcastic tweet.

4.5. Logical Integration

The predictions based on the Situational Disparity (SD) identifying features, and lexical, syntactic and pragmatic features are combined with the predictions based on the weighted-lexicon. There are two versions of this integration.

AND: Using the AND logic, a tweet is classified as sarcastic if both the predictions are sarcastic. Else it is classified as non-sarcastic.

OR: Using the OR logic, a tweet is classified as sarcastic if either of the predictions is sarcastic. Else it is classified as non-sarcastic.

5. Experimental Setup

This section describes the data that we used and the experiments conducted on the data.

5.1. Data

As discussed in Section 3, we collected 120,000 sarcastic tweets and 80,000 non-sarcastic tweets with mixed emotions (supervised by hashtags #happy, #sad, #quote). After preprocessing, we had 81,700 sarcastic tweets and 56,000 non-sarcastic tweets in our corpus.

Tweet Set-A (60,000 tweets, 30,000 sarcastic):

We randomly selected 30,000 sarcastic tweets and 30,000 non-sarcastic tweets from our corpus and removed the hashtags.

Tweet Set-B (25,000 tweets, 20,000 sarcastic):

We randomly selected 20,000 tweets from sarcastic tweets and 5,000 tweets from non-sarcastic tweets in our corpus and removed the hashtags. The same was verified by our annotators.

Tweet Set-C (20,000 tweets, 8000 sarcastic): We randomly selected 8,000 tweets from sarcastic tweets and 12,000 tweets from non-sarcastic tweets in our corpus and removed the hashtags.

Tweet Gold (2232 tweets, 505 sarcastic): This dataset was annotated for Riloff et al. (Riloff et al., 2013)[15] by 3 experts at a Cohen's statistic of $\kappa = 0.80$, $\kappa = 0.81$, $\kappa = 0.82$. The original dataset had 3000 tweets out of which 693 were positive instances of sarcasm.

Tweet Brute (139,932 tweets, 82,205 sarcastic): We combined the sarcastic and non-sarcastic tweets that we had with the Tweet Gold dataset. Now, we have 82,205 sarcastic tweets and 57,727 non-sarcastic tweets.

5.2. Sarcasm Detection Experiments

We re-implement the algorithm in Rilof et al. (Riloff et al., 2013) and compare it with our system[15]. On Tweet Set A (50% sarcastic) we extract situational disparity identifying features, from now on referred to as SD Features, and lexical, syntactic and pragmatic features as explained in sections 4.1 and 4.2. We train our classifiers for various feature combinations on LibSVM using RBF Kernel [3]. We perform a 10-fold-cross-validation (**Table 6**).

Table 6. Juxtaposing results achieved by Riloff et al and our system (part 1 of 2) on Tweet Set A.

Feature(s)	Precision	Recall	F-Score
Reimplementation of Algorithm by Riloff et al (2013)			
Contrast + Ordered	0.741	0.132	0.224
Our System (Part 1 of 2)			
SD Features	0.813	0.535	0.645
SD + Lexical	0.823	0.667	0.736
Lexical + Syntactic	0.720	0.709	0.714
Lexical + Syntactic + Pragmatic	0.831	0.844	0.837
SD + Lexical + Syntactic	0.819	0.868	0.842
SD + Lexical + Pragmatic	0.846	0.922	0.882
All features (Baseline)	**0.877**	**0.934**	**0.904**

We then performed experiments on the same dataset with the weighted-lexicon based pattern matching (**Table 7**) as discussed in Section 4.4.

Table 7. Juxtaposing results achieved by Riloff et al and our system (part 2 of 2) on Tweet Set A.

Approach	Precision	Recall	F-Score
Reimplementation of Algorithm by Riloff et al (2013)			
Contrast + Ordered	0.741	0.132	0.224
Our System (Part 2 of 2)			
Weigted-Lexicon	0.433	**0.701**	0.535

Having experimented with the two halves of our system separately, we now call in our integration system to try combinations of the two methods. We try AND and OR combinations of the predictions given by the SVM classifier based on Situational Disparity, Lexical, Syntactic and Pragmatic features, and the predictions generated through the weighted-lexicon (WL) method. (**Table 8**)

Table 8. Juxtaposing results achieved by Riloff et al and our sarcasm detection system on Tweet Set A.

Approach	Precision	Recall	F-Score
Reimplementation of Algorithm by Riloff et al (2013)			
Contrast + Ordered	0.741	0.132	0.224
Our System			
All features AND WL	0.725	0.536	0.616
All features OR WL	**0.909**	**0.973**	**0.940**

6. Evaluation

The results in Table 6 show the measures of the performance of our classifiers in terms precision, recall and f-measure on Tweet Set A. We compare our results to the algorithm given by Riloff et al which we re-implemented for evaluation purposes.

Combining the features obtained from our system as situational disparity identifiers and other lexical, syntactic and pragmatic features gave us the best measure of precision, recall and F as 0.877, 0.934, and 0.904 respectively. The f-score obtained by our system is an improvement of about 70% over the algorithm given by Riloff et al. (Riloff et al., 2013) due to the significant raise in the recall [15]. We made this our baseline to measure the reliability of the weighted-lexicon approach.

The weighted-lexicon approach (**Table 7**) individually gave an f-score of 0.535 which is a 30% improvement over that of Riloff et al. (Riloff et al., 2013) [15]. **Table 8** shows the performance of our system after both the parts are integrated. The OR integration is the best version of our system obtaining an f-score of 0.940 which is about 4% improvement over our baseline and 70% improvement over Riloff et al. (Riloff et al., 2013) [15]. We also experimented on two more sets of tweets -Tweet Set B and Tweet Set C to validate the results on different proportions of sarcastic and non-sarcastic tweets.

Table 9. Results obtained by our sarcasm detection system on Tweet Set B and Tweet Set C.

Approach	Precision	Recall	F-Score
Tweet Set B			
All features (Baseline)	0.869	0.913	0.890
Weighted-Lexicon(WL)	0.428	0.754	0.546
All features AND WL	0.725	0.439	0.546
All features OR WL	**0.911**	**0.984**	**0.9461**
Tweet Set C			
All features (Baseline)	0.874	0.909	0.891
Weighted-Lexicon(WL)	0.440	0.698	0.539
All features AND WL	0.673	0.417	0.514
All features OR WL	**0.904**	**0.938**	**0.920**

As evident from the results of our experiments on Tweet Set B and Tweet Set C (**Table 9**), our system continuously performs well with an improvement of about 5% over our baseline.

Table 10. Results obtained by our sarcasm detection system on Tweet Brute dataset.

Approach	Precision	Recall	F-Score
All features (Baseline)	0.823	0.892	0.856
All features OR WL	**0.910**	**0.961**	**0.9348**

To record the performance of the system with a larger set of data, we experimented on Tweet Brute which had 139,932 tweets. On Tweet Brute, our system obtained an improvement of 8% on f-score over our baseline, as shown in **Table 10**. Results in **Table 11** show the performance of our system on the test data Tweet Gold, gold standard test data used by Riloff et al. (Riloff et al., 2013) [15]. The f-score achieved by our system is 0.797 as compared to the best reported f-score of 0.510 by Riloff et al, outperforming the original system by about 28%. We also obtained a recall value of 0.716 which is about 27% higher than the recall of the original system.

Table 11. Comparing our system with a past work on Tweet Gold (gold standard test sample).

Approach	Precision	Recall	F-Score
Riloff et al (2013)			

Best Reported Results	0.620	0.440	0.510
Situational Disparity and Weighted-Lexicon Approach			
All Features OR WL	**0.899**	**0.716**	**0.797**

7. Conclusions

In this paper, we presented a novel approach for detecting sarcasm using situational disparity, lexical and pragmatic features, and a weighted-lexicon based pattern matching. For detecting situational disparity, we identified 4 factors: Blunt Sentiment Contrast, Word Polarity Contrast, Word-Phrase Polarity Contrast and Phrase-Phrase Polarity Contrast. The weighted-lexicon used in this paper, is a lexicon of words, patterns and phrases that cue sarcasm along with manually assigned weights juxtaposed with computationally generated confidence scores. We combined the results of statistical classifiers with those of lexicon based pattern matching using an OR logic. For experimentation, we collected 200,000 tweets from Twitter and segregated them into four sets of different proportions of sarcastic and non-sarcastic instances. We consistently obtained an improvement of at least 4% over our baseline and 70% improvement over an implementation of the algorithm described in Riloff et al (2013) on the tweet sets [15]. On testing the method on a gold standard test sample, our system outperformed Riloff et al's system by **28%** contemplating the best reported results.

Interestingly, our experiments reveal that our system performs well on large datasets with a healthy proportion of sarcastic and non-sarcastic instances. As evident, on Tweet Set A, Tweet Set B and Tweet Set C, we got F-scores of 0.94, 0.9461 and 0.92 respectively. Similarly, on Tweet Brute (139,932 tweets, 82,205 sarcastic), we obtained an F-score of 0.9348. The elated F-score can be attributed to the significant improvement in the recall.

7. References

1. J. D. Campbell and A. N. Katz. Are there necessary conditions for inducing a sense of sarcastic irony? Discourse Processes, 49(6):459–480, 2012.
2. P. Carvalho, L. Sarmento, M. J. Silva, and E. de Oliveira. Clues for detecting irony in user-generated contents: Oh...!! it's "so easy" ;-). In Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion, TSA '09, pages 53–56, New York, NY, USA, 2009. ACM.
3. C.C.Chang and C.J.Lin. Libsvm: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):27, 2011.

4. M. A. Creusere. Theories of adults' understanding and use of irony and sarcasm: Applications to and evidence from research with children. Developmental Review, 19(2):213–262, 1999.
5. D. Davidov, O. Tsur, and A. Rappoport. 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, 2010.
6. L. Ellestr¨om. Divine madness: On interpreting literature, music, and the visual arts ironically. Bucknell University Press, 2002.
7. R. J. Gerrig and Y. Goldvarg. Additive effects in the perception of sarcasm: Situational disparity and echoic mention. Metaphor and Symbol, 15:197–208, 2000.
8. K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. 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: short papers-Volume 2, pages 42–47. Association for Computational Linguistics, 2011.
9. R. Gonz´alez-Ib´anez, S. Muresan, and N. Wacholder. 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, 2011.
10. C. J. Lee and A. N. Katz. The differential role of ridicule in sarcasm and irony. Metaphor and Symbol, 13(1):1–15, 1998.
11. S. Lukin and M. Walker. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In Proceedings of the Workshop on Language Analysis in Social Media, pages 30–40, 2013.
12. D. Maynard and M. A. Greenwood. Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In Proceedings of LREC, 2014.
13. C. Nakassis and J. Snedeker. Beyond sarcasm: Intonation and context as relational cues in childrens recognition of irony. In Proceedings of the Twenty-Sixth Boston University Conference on Language Development. Cascadilla Press, Somerville, MA, pages 429–440, 2002.

14. J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth. The development and psychometric properties of LIWC2007, 2007.
15. E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang. Sarcasm as contrast between a positive sentiment and negative situation. In Proceedings of EMNLP., pages 701–714, 2013.
16. J. Schwoebel, S. Dews, E. Winner, and K. Srinivas. Obligatory processing of the literal meaning of ironic utterances: Further evidence. Metaphor and Symbol, 15:47–61, 2000.
17. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2):267–307, 2011.
18. J. Tepperman, D. R. Traum, and S. Narayanan. " yeah right": sarcasm recognition for spoken dialogue systems. In INTERSPEECH. Citeseer, 2006.
19. O. Tsur, D. Davidov, and A. Rappoport. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In ICWSM, 2010.