Sarcasm Detection on Twitter: A Behavioral Modeling Approach

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu
Computer Science and Engineering
Arizona State University
{arajades, reza, huan.liu}@asu.edu

ABSTRACT

Sarcasm is a nuanced form of language in which individuals state the opposite of what is implied. With this intentional ambiguity, sarcasm detection has always been a challenging task, even for humans. Current approaches to automatic sarcasm detection rely primarily on lexical and linguistic cues. This paper aims to address the difficult task of sarcasm detection on Twitter by leveraging behavioral traits intrinsic to users expressing sarcasm. We identify such traits using the user's past tweets. We employ theories from behavioral and psychological studies to construct a behavioral modeling framework tuned for detecting sarcasm. We evaluate our framework and demonstrate its efficiency in identifying sarcastic tweets.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

Sarcasm Detection; Behavioral Modeling; Social Media

1. INTRODUCTION

In recent years, social media sites such as Twitter have gained immense popularity and importance. These sites have evolved into large ecosystems where users express their ideas and opinions uninhibitedly. Companies leverage this unique ecosystem to tap into public opinion on their products or services and to provide real-time customer assistance. Not surprisingly, most large companies have a social media presence and a dedicated team for marketing, after-sales service, and consumer assistance through social media.

With the high velocity and volume of social media data, companies rely on tools such as HootSuite[1], to analyze data and to provide customer service. These tools perform tasks such as content management, sentiment analysis, and extraction of relevant messages for the company's customer service representatives to respond to. However, these tools lack the sophistication to decipher more nuanced forms of language such as sarcasm or humor, in which the meaning of a message is not always obvious and explicit. This imposes an extra burden on the social media team — already inundated with customer messages — to identify these messages and respond appropriately. Table 1 provides two examples where the customer service representatives fail to detect sarcasm. Such public gaffes not only upset the already disgruntled customers but also ruin the public images of companies.

Our goal in this study is to tackle the difficult problem of sarcasm detection on Twitter. While sarcasm detection is inherently challenging, the style and nature of content on Twitter further complicate the process. Compared to other, more conventional sources such as news articles and novels, Twitter is (1) more informal in nature with an evolving vocabulary of slang words and abbreviations and (2) has a limit of 140 characters per tweet which provides fewer word-level cues and adds more ambiguity.

Current research on sarcasm detection on Twitter [38, 10, 20, 29] has primarily focused on obtaining information from the text of the tweets. These techniques treat sarcasm as a linguistic phenomenon, with limited emphasis on the psychological aspects of sarcasm. However, sarcasm has been extensively studied in psychological and behavioral sciences and theories explaining when, why, and how sarcasm is expressed have been established. These theories can be extended and employed to automatically detect sarcasm on Twitter. For example, Rockwell [32] identified a positive correlation between cognitive complexity and the ability to produce sarcasm. A high cognitive complexity of an individual can be manifested on Twitter in terms of the language complexity of the tweets.

Hence, to follow a systematic approach, we first theorize the core forms of sarcasm using existing psychological and behavioral studies. Next, we develop computational features to capture these forms of sarcasm using user's current and past tweets. Finally, we combine these features to train a learning algorithm to detect sarcasm. We make the following contributions in this paper:

1. We identify different forms of sarcasm and demonstrate how these forms are manifested on Twitter.

2. We introduce behavioral modeling as a new, effective approach for detecting sarcasm on Twitter; we pro-

[1]https://hootsuite.com/

WSDM'15, February 2–6, 2015, Shanghai, China.

ACM 978-1-4503-3317-7/15/02 ...$15.00.
http://dx.doi.org/10.1145/2684822.2685316.

Table 1: Examples of Misinterpreted Sarcastic Tweets.

Example	Users	Tweets
1	User 1	you are doing great! Who could predict heavy travel between #Thanksgiving and #NewYearsEve. And bad cold weather in Dec! Crazy!
	Major U.S Airline	We #love the kind words! Thanks so much.
	User 1	wow, just wow, I guess I should have #sarcasm
2	User 2	Ahhh..**** reps. Just had a stellar experience w them at Westchester, NY last week. #CustomerSvcFail
	Major U.S Airline	Thanks for the shout-out Bonnie. We're happy to hear you had a #stellar experience flying with us. Have a great day.
	User 2	You misinterpreted my dripping sarcasm. My experience at Westchester was 1 of the worst I've had with ****. And there are many.

pose and evaluate the **SCUBA** framework — **S**arcasm **C**lassification **U**sing a **B**ehavioral modeling **A**pproach.

3. We investigate and demonstrate the importance of historical information available from past tweets for sarcasm detection.

In section 2, we review related sarcasm detection research. In section 3, we formally define sarcasm detection on Twitter. In section 4, we discuss different forms of sarcasm and outline SCUBA, our behavioral modeling framework for detecting sarcasm. In section 5, we demonstrate how different forms of sarcasm can be identified within Twitter. In section 6, we detail our experiments. Section 7 concludes this research with directions for future work.

2. RELATED WORK

Sarcasm has been widely studied by psychologists, behavioral scientists and linguists for many years. Theories explaining the cognitive processes behind sarcasm usage such as the echoic reminder theory [18], allusional pretense theory [19], and implicit display theory [39] have been extensively researched. However, automatic detection of sarcasm is a relatively unexplored research topic and a challenging problem [25]. While studies on automatic detection of sarcasm in speech [35] utilizes prosodic, spectral and contextual features, sarcasm detection in text has relied on identifying text patterns [4] and lexical features [10, 17].

Davidov et al. [4] devised a semi-supervised technique to detect sarcasm in Amazon product reviews and tweets. They used interesting pattern-based (high frequency words and content words) and punctuation-based features to build a weighted k-nearest neighbor classification model to perform sarcasm detection. Reyes et al. [28] focused on developing classifiers to detect verbal irony based on ambiguity, polarity, unexpectedness and emotional cues derived from text. González-Ibáñez et al. [10] introduced a sarcasm detection technique using numerous lexical features (derived from LWIC [27] and Wordnet Affect [34]) and pragmatic features such as emoticons and replies. Liebrecht et al. [20] used unigrams, bigrams and trigrams as features to detect sarcastic Dutch tweets using a balanced winnow classifier. More recently, Riloff et al. [29], used a well-constructed lexicon-based approach to detect sarcasm based on an assumption that sarcastic tweets are a contrast between a positive sentiment and a negative situation.

As described, past studies on sarcasm detection have primarily focused on linguistic aspects of sarcasm and used only the text of the tweet. We introduce a systematic approach for effective sarcasm detection by not only analyzing the content of the tweets but by also exploiting the behavioral traits of users derived from their past activities. We map research on (1) *what causes people to use sarcasm?*, (2) *when is sarcasm used?* and (3) *how is sarcasm used?*, to observable user behavioral patterns on Twitter that can help build a comprehensive supervised framework to detect sarcasm.

3. PROBLEM STATEMENT

Sarcasm, while similar to irony, differs in that it is usually viewed as being caustic and derisive. Some researchers even consider it to be aggressive humor [1] and a form of verbal aggression [37]. While researchers in linguistics and psychology debate about what exactly constitutes sarcasm, for the sake of clarity, we use the Oxford dictionary's definition of sarcasm[2] as "*a way of using words that are the opposite of what you mean in order to be unpleasant to somebody or to make fun of them.*" We formally define the sarcasm detection problem on Twitter as follows:

Definition. ***Sarcasm Detection on Twitter.*** *Given an unlabeled tweet* $\boldsymbol{t}$ *from user* $\mathbf{U}$ *along with a set of* $\mathbf{U}$*'s past tweets* $\mathbf{T}$*, a solution to sarcasm detection aims to automatically detect if* $\boldsymbol{t}$ *is sarcastic or not.*

In addition to following a behavioral modeling approach, our problem is different from past sarcasm detection research which use only text information from $\boldsymbol{t}$ and do not consider the user's past tweets $\mathbf{T}$ that are available on Twitter.

4. SCUBA: BEHAVIORAL MODELING FRAMEWORK

Sarcastic tweets are not always created in isolation. When posting sarcastic tweets, users make conscious efforts to express their thoughts through sarcasm. They may decide to use sarcasm as a behavioral response to a certain situation, observation, or emotion. These situations, observations, or emotions may be observed and analyzed on Twitter.

It is observed that some individuals have more difficulty in creating or recognizing sarcasm than others due to cul-

[2]www.oxfordlearnersdictionaries.com/definition/english/sarcasm

tural differences, language barriers, and the like. In contrast, some individuals have a higher propensity to use sarcasm than others. Hence, SCUBA also considers the user's likelihood of being a *sarcastic person*. This can be achieved on Twitter by analyzing the user's past tweets. Using existing research on sarcasm and our observations on Twitter, we find that sarcasm generation can be characterized as one (or a combination) of the following:

Sarcasm as a contrast of sentiments

A popular perception of sarcasm among researchers is that sarcasm is a contrast of sentiments. A classical view of sarcasm, based on the traditional pragmatic model [11], argues that sarcastic utterances are first processed in the literal sense and if the literal sense is found incompatible with the present context, only then is the sentence processed in its opposite (ironic) form. This perceived contrast may be expressed with respect to mood, affect or sentiment.

Sarcasm as a complex form of expression

Rockwell [32] showed that there is a small but significant correlation between cognitive complexity and the ability to produce sarcasm. A high cognitive complexity involves understanding and taking into account, multiple perspectives to make cogent decisions. Furthermore, expressing sarcasm requires determining if the environment is suitable for sarcasm, creating an appropriate sarcastic phrase and assessing if the receiver would be capable of recognizing sarcasm. Therefore, sarcasm is a complex form of expression needing more effort than usual from the user.

Sarcasm as a means of conveying emotion

Sarcasm is primarily a form of conveying one's emotions. While sarcasm is sometime interpreted as aggressive humor [1] or verbal aggression [37], it also functions as a tool for self expression. Past studies [12], recognize that sarcasm is usually expressed in situations with negative emotions and attitudes.

Sarcasm as a possible function of familiarity

Friends and relatives are found to be better at recognizing sarcasm than strangers [31]. Further, it has been demonstrated that the knowledge of language [3] and culture [33] also play an important role in the recognition and usage of sarcasm.

Sarcasm as a form of written expression

Sarcasm in psychology has been studied primarily as a spoken form of expression. However, sarcasm is quite prevalent in written form as well, especially with the advent of online social networking sites. Through time, users have become more adept at conveying sarcasm in writing by including subtle markers that indicate to the unassuming reader, that the phrase might be sarcastic. For example, while "*you're so smart*" does not hint at sarcasm, "*Woowwww you are SOOOO cool*"[3] elicits some doubts about the statement's sincerity.

[3]An original tweet collected.

We believe that when expressing sarcasm, the user would invariably exhibit one or more of these forms. Therefore, SCUBA incorporates a behavioral modeling approach [42] for sarcasm detection that utilizes features which capture the different forms of sarcasm. These extracted features are utilized in a supervised learning framework along with some labeled data to determine if the tweet is sarcastic or not. In our setting, labeled data is a set of tweets, among which sarcastic tweets are known. As the novelty of the approach lies in the behavioral modeling and not the actual classifier, we explain in detail how sarcasm can be modeled and incorporated into SCUBA in the following section.

5. SCUBA: REPRESENTING FORMS OF SARCASM

Users' efforts in generating sarcasm are manifested in many ways on Twitter. In this section, we describe how the aforementioned forms are realized on Twitter and how one can construct relevant features to capture these form in the context of Twitter.

5.1 Sarcasm as a contrast of sentiments

5.1.1 Contrasting connotations

A common means of expressing sarcasm is to use words with contrasting connotations within the same tweet. For example, in *I love getting spam emails!*, *spam* obviously has a negative connotation while *love* is overwhelmingly positive. To model such occurrences, we construct features based on (1) affect and (2) sentiment scores.

We obtain affect score of **words** from a dataset compiled by Warriner et al. [41]. This dataset contains affect (valence) scores for 13,915 English lemmas which are on a 9-point scale, with 1 being the least pleasant.

The sentiment score is calculated using SentiStrength [36]. SentiStrength is a lexicon-based tool optimized for tweet sentiment detection based on sentiments of individual words in the tweet. Apart from providing a ternary sentiment result {positive, negative, neutral} for the whole tweet, SentiStrength outputs two scores for each word. A negative sentiment score from -1 to -5 (not-negative to extremely-negative) and a positive sentiment score from 1 to 5 (not-positive to extremely-positive). Here, we use SentiStrength's lexicon to obtain **word** sentiment scores. From these sentiment and affect scores, we compute the following:

$$A = \{\, \text{affect}(w) \mid w \,\epsilon\, t\}, \tag{1}$$

$$S = \{\, \text{sentiment}(w) \mid w \,\epsilon\, t\}, \tag{2}$$

$$\Delta_{\text{affect}} = \max(A) - \min(A), \tag{3}$$

$$\Delta_{\text{sentiment}} = \max(S) - \min(S), \tag{4}$$

where t is the tweet and w is a word in t. The affect(w) outputs the affect score of word w. The sentiment(w) outputs the sentiment score of word w. Δ_{affect} and $\Delta_{\text{sentiment}}$ indicate the level of contrast in terms of affect and sentiment infused into the tweet by the user. We use Δ_{affect} and $\Delta_{\text{sentiment}}$ as features (2 features).

SentiStrength and the dataset provided by Warriner et al. [41] can only provide sentiment and affect scores for unigrams. Hence, we construct a lexicon of positive and negative sentiment bigrams and trigrams used on Twitter following an approach similar to Kouloumpis et al. [16] as follows:

1. We collect about 400,000 tweets with positive sentiment hashtags such as `#love`, `#happy`, `#amazing`, etc., and 400,000 tweets with negative sentiment hashtags such as `#sad`, `#depressed`, `#hate`, among others.

2. From these tweets, we extracted the bigrams and trigrams along with their respective frequencies. We filter out bigrams and trigrams with frequencies less than 10.

3. For each bigram or trigram b, we find its associated sentiment score, $\frac{POS(b) - NEG(b)}{POS(b) + NEG(b)}$, where $POS(b)$ is the number of occurrences of b in the positive tweets dataset and $NEG(b)$ is the number of occurrences of b in the negative tweets dataset. We filter out bigrams or trigrams with sentiment scores $\in (-0.1, 0.1)$. This sentiment measure is similar to association scores given by Liu et al. [21].

Using the generated lexicon, we include as features, the number of n-grams with positive sentiment scores, the number of n-grams with negative sentiment scores, the summation of scores for positive n-grams, and the summation of scores for negative n-grams (4 features).

5.1.2 Contrasting present with the past

Sometimes, the user may set up a contrasting context in her previous tweet and then, choose to use a sarcastic remark in her current tweet. To model such behavior, we obtain the sentiment expressed by the user (i.e., positive, negative, neutral) in the previous tweet and the current tweet using SentiStrength. Then, we include the type of sentiment transition taking place from the past tweet to the current tweet (for example, *positive* → *negative*, *negative* → *positive*) as a feature (1 feature). In total, there are nine such transitions involving the combinations of positive, negative and neutral sentiments. To provide a historical perspective on the user's likelihood for such sentiment transitions, we compute the probability for all nine transitions using the user's past tweets. The transition probabilities along with the probability of the current transition are included as features in our framework (10 features).

5.2 Sarcasm as a complex form of expression

5.2.1 Readability

As sarcasm is widely acknowledged to be hard to read and understand, we adapt standardized readability tests to measure the degree of complexity and understandability of the tweet. We use as features: number of words, number of syllables, number of syllables per word in the tweet derived from the Flesch-Kincaid Grade Level Formula [8], number of polysyllables[4] and the number of polysyllables per word in the tweet derived from SMOG [22] (6 features).

Inspired by the average word length feature used in the Automated Readability Index [15], we formulate a more comprehensive set of features using the word length distribution $L = \{l_i\}_{i=1}^{19}$ constructed from tweet t as follows:

1. For each word w in t, we compute its character length $|w|$. For convenience, we ignore words of length 20 or more. We construct a word length distribution $L = \{l_i\}_{i=1}^{19}$ for t, where l_i denotes the number of words in the tweet with character length i.

2. L may be represented succinctly using the following 6-tuple presentation:

$$< \mathbb{E}[l_w], med[l_w], mode[l_w], \sigma[l_w], \min_{w \in t} l_w, \max_{w \in t} l_w >, \quad (5)$$

where $\mathbb{E}$ is the mean, *med* is the median, *mode* is the mode and σ is the standard deviation.

We include the 6-tuple representation as features in our framework (6 features).

Further, given the availability of the user's past tweets, we examine if there is a noticeable difference in the word length distribution between the user's current tweet and her past tweets. It must be noted that while sarcastic tweets may also be present in the user's past tweets, because of their relative rarity, the past tweets when taken in entirety, would 'average out' any influence possibly introduced by a few past sarcastic tweets. Therefore, any difference from the norm in the word length distribution of the current tweet can be captured. To capture differences in word length distribution, we perform the following steps:

1. From the user's current tweet, we construct a probability distribution D_1 over length of words in the tweet.

2. From the user's past tweets, we construct a probability distribution D_2 over length of words in all the past tweets.

3. To calculate the difference between the world length distribution of the current tweet and the past tweets, we calculate the Jensen-Shannon (JS) divergence between D_1 and D_2:

$$JS(D_1||D_2) = \tfrac{1}{2}KL(D_1||M) + \tfrac{1}{2}KL(D_2||M), \quad (6)$$

where $M = \frac{D_1 + D_2}{2}$ and KL is the KL-divergence:

$$KL(T_1||T_2) = \sum_i \ln(\tfrac{T_1(i)}{T_2(i)})T_1(i).$$

We include the JS-divergence value as a feature (1 feature).

5.3 Sarcasm as a means of conveying emotion

5.3.1 Mood

Mood represents the user's state of emotion. Intuitively, the mood of the user may be indicative of her propensity to use sarcasm; if the user is in a bad (negative) mood, she may choose to express it in the form of a sarcastic tweet. Therefore, we gauge the user's mood using sentiment expressed in her past tweets. However, we cannot assume that the user's mood is encapsulated in her last n tweets. Therefore, we capture the mood using her past tweets as follows:

1. For each past tweet t, we compute its positive sentiment score pos(t) and its absolute negative sentiment score neg(t) using SentiStrength.

2. We divide the user's past tweets into overlapping buckets based on the number of tweets posted prior to the current tweet.

3. Each bucket b_n consists of the previous n tweets posted by the user. We select $n \in \{1, 2, 5, 10, 20, 40, 80\}$.

[4]Polysyllables are words containing three or more syllables.

4. In each b_n, we capture the user's perceived mood using two tuples. The first tuple consists of four features:

$$< \sum^{+}, \sum^{-}, P, \max(\sum^{+}, \sum^{-}) >, \quad (7)$$

where $\sum^{+}$ and $\sum^{-}$ are the total positive and negative sentiments in b_n: $\sum^{+} = \sum_{t \in b_n} \text{pos}(t)$, $\sum^{-} = \sum_{t \in b_n} \text{neg}(t)$. P is either $+$ or $-$. $P = +$, when $\sum^{+} > \sum^{-}$, and $P = -$, otherwise. The second tuple consists of six features:

$$< n_+, n_-, n_0, n, Q, \max(n_+, n_-, n_0) >, \quad (8)$$

where n_+ is the number of positive tweets, n_- is the number of negative tweets, and n_0 is the number of neutral tweets present in b_n(found using SentiStrength). n is the total tweets present in b_n and Q indicates what the majority of tweets are, i.e., $Q \in \{+, -, 0\}$. For example, $Q = +$, when $n_+ = max(n_+, n_-, n_0)$. We include both tuples for each b_n as features in SCUBA ($7 \times (4 + 6) = 70$ features).

As one's mood remains constant for a limited amount of time, we also gauge the user's mood within a specific time window. However, we cannot assume that the user's mood is encapsulated within t minutes. Therefore, we divide the user's past tweets into buckets b_t, which consists of all the tweets posted by the user within t minutes from the current tweet. Here, $t \in \{1, 2, 5, 10, 20, 60, 720, 1440\}$ minutes (1440 minutes = 1 day). For each bucket b_t, we include the tuples in (7) and (8) also as features ($8 \times 10 = 80$ features).

5.3.2 Affect and sentiment

As sarcasm is a combination of affect and sentiment expression, we examine how affect and sentiment are expressed in sarcastic tweets. To this end, we construct a sentiment score distribution SD. SD consists of 11 values, each value being the number of words in the tweet with sentiment score i, where $i \in [-5, 5]$. We also construct an affect score distribution AD. AD contains 9 values. Each value is the number of words in the tweet with an affect score j, where $j \in [1, 9]$. We normalize counts in SD and AD. We include both distributions as features (11+9=20 features). Similar to Eq. (5), we represent these distributions as 6-tuples and include them as features (12 features). We also include the number of affect words, number of sentiment words, and the tweet's overall sentiment (positive, negative, or neutral) as features in SCUBA (3 features).

To capture differences in sentiment expression in sarcastic tweets versus non-sarcastic ones, we compare the sentiment score distribution of the user's past tweets to that of her current tweet. Following a procedure similar to that of Section 5.2.1, we calculate the JS-divergence value between the past and current sentiment score distributions and include it as a feature (1 feature).

Finally, to gain insights into how a user employs Twitter to express emotion, we determine the range of sentiments expressed by the user in the past. To perform this, for all sentiment scores $i \in [-5, 5]$, we compute the number of tweets with sentiment score i in all past tweets of the user. By normalizing these counts, we obtain a probability distribution over sentiment scores in past tweets. We include this probability distribution as a feature (11 features).

5.3.3 Frustration

When individuals experience unjust situations, they sometimes turn to social media as an effective outlet for their complaints and frustrations [2]. These frustrations are often expressed in the form of sarcasm [9] (see example tweets in Table 1). We hypothesize that as users encounter unpleasant situations in real life, they react spontaneously by posting tweets to vent out their frustration. Therefore, they diverge from their regular tweeting patterns.

To capture this behavioral pattern, using the user's past tweets, we construct an expected tweet posting time probability distribution. From each of the user's past tweets, we extract the tweet creation time, using which, we build a normalized 24 bin distribution TD (one for each hour). TD approximates the probability of the user tweeting at each hour. For each tweet, using the TD for the user posting it, we find the likelihood of the user posting the tweet at that hour. The lower the likelihood, the more divergent the tweet is from the user's usual tweeting patterns. Low likelihood scores indicate that the user is not expected to tweet at that particular time and that the user has gone out of her way to tweet at that time, therefore, in some sense, the tweet is spontaneous in nature. We include the likelihood of the user tweeting at that particular hour as a feature (1 feature).

We also observe that users tend to post successive tweets in short quick bursts when they vent out their frustrations; therefore, we include the time difference between the examined tweet and the previous tweet posted by the user as a feature (1 feature). Finally, a common way to express frustration is by using swear words. Using the list of most common swear words provided by Wang et al. [40], we check for the presence of such words in the tweet and include their presence using a boolean feature (1 feature).

5.4 Sarcasm as a possible function of familiarity

5.4.1 Familiarity of language

Intuitively, one would expect a user who uses a form of language as complex as sarcasm to have good command over the language. Therefore, we measure the user's language skills with features that are inspired by standardized language proficiency cloze tests. In cloze tests, proficiency is evaluated based on vocabulary, grammar, dictation, and reading levels [23]. As dictation and reading levels pertain to the oratory and reading skills of the user that cannot be measured from written text, we focus on constructing features that best represent the vocabulary and grammar skills.

Vocabulary Skills. We determine the size of user's vocabulary from user's past tweets. We include as features, the total number of words, total number of distinct words used and the ratio of distinct words to total words used to measure the user's redundancy in word usage (3 features).

Grammar Skills. To measure grammar skills, we investigate how the user employs different parts-of-speech (POS). The POS tags for words in the tweets are generated using TweetNLP's [24] POS tagger. TweetNLP generates 25 possible tags such as *interjections* or *emoticons*. We obtain the POS tag for every word in the tweet and build a corresponding POS probability distribution and include it as features (25 features). As English grammar is intricate and nuanced, it is difficult to extensively measure a user's grammar exper-

tise. However, one can check for correct grammatical usage for commonly used words. We check the correct grammatical usage for "your" and "its", both frequently used. We observe that users often mistakenly use words such as "your" instead of "you're' and "its" instead of "it's". Using all past tweets of a user, we obtain the POS of the word used immediately after "your" and "its". If the word has been used in the correct grammatical sense, the POS of the succeeding word should not be a verb (example, "your doing great!" is incorrect), adverb (example, "its freakin' amazing" is incorrect) or a determiner (such as "a" or "the"). We include as features, the fraction of times the two words were used in incorrect grammatical form by the user (2 features). There are other POS that can render the usage of "your" or "its" incorrect; however, for correctness, we adopt a conservative strategy by checking only for verbs, adverbs, and determiners.

Sarcastic users, in addition to their vocabulary and grammar skills, are familiar with sarcasm as an expression form.

Familiarity with Sarcasm. For measuring user's familiarity with sarcasm, we include the number of past occurrences of `#not` or `#sarcasm` hashtags as a feature (1 feature). Further, it has been shown that people in different regions perceive and use sarcasm differently (see Dress et al.'s study [5]). Thus, we try to infer the location of the user. However, the user provided location on Twitter is often noisy as it is a *free-text* field in which any text may be inputted. Therefore, we approximate the user's location with her time zone and include it as a feature (1 feature).

5.4.2 Familiarity of environment

Users express sarcasm better when they are well acquainted with their environment. Just as people are less likely to use sarcasm at a new, unfamiliar setting, users take time to get familiar with Twitter before posting sarcastic tweets. We measure a user's familiarity with Twitter in terms of her usage familiarity, parlance familiarity, and social activity.

Usage Familiarity. We measure usage familiarity using the number of tweets posted, number of days on Twitter (i.e., Twitter age), and the average number of daily tweets and include all as features (3 features). These features provide indications of the duration and the intensity at which the user has been using Twitter. We also measure familiarity in terms of the user's frequency of Twitter usage. From the user's past tweets, we compute the time differences between all pairs of successive tweets. We represent the distribution of these time differences as a 6-tuple similar to Eq. (5), and include them as features (6 features).

Twitter Parlance Familiarity. To capture user familiarity with *Twitter parlance*, we include the number of retweets, mentions and hashtags used in past tweets as features (3 features). Experienced Twitter users often use shortened words (by removing vowels, using numbers, etc.) to circumvent the 140 character limit. Hence, we include the presence of alphanumeric words (boolean), presence of words without vowels (boolean), as well as the percentage of dictionary words present in the tweet as features (3 features).

Social Familiarity. We measure social familiarity by identifying how embedded a user is in Twitter's social graph. Hence, we include the number of friends and followers as features (2 features). To adjust for longevity, we divide the numbers of friends and followers by the user's Twitter age and include them as features (2 features).

5.5 Sarcasm as a form of written expression

While low pitch, high intensity and a slow tempo [30] are vocal indicators of sarcasm, users attempting to express sarcasm in writing are devoid of such devices. Therefore, users may be forced to use certain writing styles to compensate for the lack of visual or verbal cues. We categorize such behavior into prosodic and structural variations.

5.5.1 Prosodic variations

Users often repeat letters in words to stress and overemphasize certain parts of the tweet (for example, *sooooo*, *awesomeeee*) to indicate that they mean the opposite of what is written. We capture such usage by including as boolean features, the presence of repeated characters (3 or more) and the presence of repeated characters (3 or more) in sentiment-loaded words (such as, loveeee) (2 features). We also include the number of characters used, and the ratio of the number of distinct characters to the total characters used in the tweet as features (2 features).

We also observe that users often capitalize certain words to emphasize changes in tone (if the tweet were to be read out loud). We account for such changes by including the number of capitalized words in the tweet as a feature (1 feature). Other users capitalize certain parts-of-speech (POS) to exaggerate or to vent their frustration. Using TweetNLP, we obtain the POS tag for each capitalized word in the tweet. Then, we compute the probability distribution of POS tags for capitalized words and include it as features (25 features).

Users also use certain punctuations to express non-verbal cues that are crucial for sarcasm deliverance in speech. For example, users use "*" to indicate emphasis, "..." to indicate pause, "!!!" for exclamations (sometimes over-done to indicate sarcasm). Thus, we include as features, the normalized distribution of common punctuation marks (.,!?'*") (7 features). To compare the user's current usage of punctuations to her past usage, similar to Eq. (6), we calculate the JS-divergence value between the current and past punctuation distribution and include it as a feature (1 feature).

5.5.2 Structural variations

We observe that sarcastic tweets sometimes have a certain structure wherein the user's views are expressed in the first few words of the tweet, while in the later parts, a description of a particular scenario is put forth, e.g., *I love it when my friends ignore me*. To capture possible syntactic idiosyncrasies arising from such tweet construction, we use as features, the POS tags of the first three words and the last three words in the tweet (6 features). We also include the position of the first sentiment-loaded word (0 if not present) and the first affect-loaded word (0 if not present) as a feature (2 features). Given the structure followed in constructing sarcastic tweets, we also check for positional variations in the hashtags present in the tweet. We trisect the tweet based on the number of words present and include as features the number of hashtags present in the each of the three parts of the tweet (3 features).

To capture differences in syntactic structures, we examine parts of speech tags present in the tweet. Similar to Eq. (6), we construct a probability distribution over the POS tags present in the current tweet as well as POS tags in past tweets and include the Jensen-Shannon divergence value between the two distribution as a feature (1 feature).

Past studies on quantifying linguistic style [14] have used lexical density, intensifiers, and personal pronouns as important measures to gauge the writing style of the user. Lexical density is the fraction of information carrying words present in the tweet (nouns, verbs, adjectives, and adverbs). Intensifiers are words that maximize the effect of adverbs or adjectives (for example, so, or very). Personal pronouns are pronouns denoting a person or group (for example, me, our, or her). We include as features the lexical density, the number of intensifiers used, the number of first-person singular, first-person plural, second-person, and third-person pronouns present in the tweet (6 features).

In total, we construct 335 features based on the behavioral aspects of sarcasm.

6. EXPERIMENTS

In this section, the SCUBA framework is systematically evaluated through a series of experiments. First, we detail the data collection process and our dataset. Next, we train SCUBA and compare its performance to baselines. Then, we determine the contribution of different feature sets to SCUBA's performance. We conduct feature importance analysis to determine a small set of features that are most beneficial for sarcasm detection. Finally, we examine the robustness of our framework under different scenarios.

6.1 Data collection

We validate our framework using a dataset[5] of tweets from Twitter. To obtain a set of sarcastic tweets, we query the Streaming API using keywords `#sarcasm` and `#not` filtering out non-English tweets and retweets. We also remove tweets containing mentions and URLs as obtaining information from media and URLs is computationally expensive. We limit our analysis to tweets which contain more than three words as we found that tweets with fewer words were very noisy or clichéd (e.g., *yeah, right! #sarcasm*). Davidov et al. [4] noted that some tweets containing the `#sarcasm` hashtag were *about* sarcasm and that the tweets themselves were not sarcastic. To limit such occurrences, we include only tweets that have either of the two hashtags as its last word; this reduces the chance of obtaining tweets that are not sarcastic. After preprocessing, we obtained about 9104 sarcastic tweets which were self described by users as being sarcastic using the appropriate hashtags. We remove the `#sarcasm` and `#not` hashtags from the tweets before proceeding with the evaluation.

To collect a set of general tweets (not sarcastic), we used Twitter's Sample API which provides a random sample of tweets. These tweets were subjected to the same aforementioned preprocessing technique. Finally, for each tweet in the collected dataset, we extract the user who posted the tweet and then, we obtained that user's past tweets (we obtain the past 80 tweets for each user).

Some examples of tweets in the dataset are:

1. *This paper is coming along... #not*
2. *Finding out your friends' lives through tweets is really the greatest feeling. #sarcasm*

The above examples illustrate the difficulty of the task at hand. The first tweet may or may not be sarcastic purely depending on the context (which is not available in the tweet). Even if some background is provided, as in the case of the second tweet, clearly, it is still a complicated task to map that information to sarcasm.

It must also be noted that, to avoid confusion and ambiguity when expressing sarcasm in writing, the users choose to explicitly mark the sarcastic tweets with appropriate hashtags. The expectation is that these tweets, if devoid of these hashtags, might be difficult to comprehend as sarcasm even for humans. Therefore, our dataset might be biased towards the hardest forms of sarcasm. Using this dataset, we evaluate our framework and compare it with existing baselines.

[5]The dataset can be obtained by contacting the first author.

6.2 Performance evaluation

Naturally, the class distribution over tweets is skewed towards the non-sarcastic tweets. Therefore, we evaluate the SCUBA framework using different class distributions (1:1, 10:90, 20:80, where 1:1 means for every sarcastic tweet in the dataset, we introduce 1 tweet that is not sarcastic.). We include AUC (Area under the ROC Curve) apart from accuracy as a performance measure as AUC is robust to class imbalances [7]. This analysis gives an insight into how well SCUBA performs under varied distributions. We compare SCUBA to the following baselines.

6.2.1 Baselines

We compare our framework against 6 baselines. The first two are based on a state-of-the-art lexicon-based technique by Riloff et al. [29]. The basic premise of their method is that sarcasm can be viewed as a contrast between a positive sentiment and a negative situation. They construct three phrase lists (positive verb phrases, positive predicative expressions, and negative situations) from 175,000 tweets using a parts-of-speech aware bootstrapping technique that extracts relevant phrases. Different combinations of these phrase lists were used to decide if a tweet is sarcastic or not. Using their phrase lists, we re-implement two of their approaches:

[1] Contrast Approach. The method marks a tweet as sarcastic if it contains a positive verb phrase or positive predicative expression along with a negative situation phrase.

[2] Hybrid Approach. The method marks a tweet as sarcastic if it is marked sarcastic either by the bootstrapped-lexicon approach or by a bag-of-words classifier trained on unigrams, bigrams, and trigrams. To provide a comparable framework to the Hybrid Approach, we include the prediction of the discussed n-gram classifier into SCUBA as a feature. We call our n-gram augmented framework, SCUBA++.

[3] SCUBA - `#sarcasm`. To quell doubts that SCUBA merely labels all tweets from users who have previously used `#sarcasm` or `#not` as sarcastic, we completely remove that particular feature and perform the same classification task.

[4] Random Classifier. The baseline classifies the tweets randomly into sarcastic and non-sarcastic.

[5] Majority Classifier. It classifies all tweets into the majority class (known from the class distribution).

[6] n-gram Classifier. The same n-gram model used in the hybrid approach that classifies tweets into sarcastic and non-sarcastic based on their unigrams, bi-grams, and tri-grams.

Table 2: Performance Evaluation using 10-fold Cross-Validation.

Technique	Dataset Distribution					
	1:1		20:80		10:90	
	Acc.	AUC	Acc.	AUC	Acc.	AUC
SCUBA	83.46	0.83	88.10	0.76	92.24	0.60
Contrast Approach	56.50	0.56	78.98	0.57	86.59	0.57
SCUBA++	86.08	0.86	89.81	0.80	92.94	0.70
Hybrid Approach	77.26	0.77	78.40	0.75	83.87	0.67
SCUBA - `#sarcasm`	83.41	0.83	87.53	0.74	91.87	0.63
n-gram Classifier	78.56	0.78	81.63	0.76	87.89	0.65
Majority Classifier	50.00	0.50	80.00	0.50	90.00	0.50
Random Classifier	49.17	0.50	50.41	0.50	49.78	0.50

Table 3: Feature Set Analysis.

Features	Accuracy
All features	83.46 %
− Complexity-based features	73.00 %
− Contrast-based features	57.34 %
− Emotion expression-based features	71.52 %
− Familiarity-based features	73.67 %
− Text expression-based features	76.72 %

6.2.2 *Training the Framework*

Before training, we select a suitable classifier for SCUBA. We evaluate SCUBA's performance using multiple supervised learning algorithms on our collected dataset (with class distribution 1:1). We evaluate using a J48 decision tree, ℓ_1-regularized logistic regression, and ℓ_1-regularized ℓ_2-loss SVM[6] to obtain an accuracy of 78.06%, 83.46%, and 83.05%, respectively. We choose ℓ_1-regularized logistic regression for comparison with the baselines.

We use 10-fold cross-validation technique to evaluate the framework's performance, the results of which are given in Table 2. From the results, we observe that SCUBA++ clearly outperforms all other techniques for every class distribution and for both accuracy and AUC. Note that only SCUBA and SCUBA++ perform better than the majority classifier for highly skewed distributions (90:10). We also observe that while the Hybrid Approach performs much better than the Contrast Approach, it is still not very effective for skewed distributions. Also, we notice that when the past sarcasm feature is removed from SCUBA, we obtain similar performance outcomes indicating the minimal effect of using this feature on the framework's performance. Both random classifier and the majority classifier obtain an AUC score of 0.5, which is the minimum possible AUC score attainable. To evaluate the benefit of using different feature sets, we perform the following feature set analysis. This analysis allows us to make informed decisions about which feature sets to consider if computationally constrained.

6.3 Feature set analysis

We divide the list features into sets depending on the different forms of sarcasm from which they were derived - features based on complexity, based on contrast, based on expression of emotion, based on familiarity, and based on expression in text form.

[6]We use Weka [13], LIBLINEAR [6], and Scikit-learn [26].

Table 3 shows the performance of SCUBA using each of the feature sets individually. While all feature sets contribute to SCUBA's performance, they do so unequally. Clearly, all feature sets perform much better than contrast-based features. This further shows the need to view sarcasm through its varied facets and not a particular form of expression (such as contrast seeking).

To gain deeper insights into which specific features are most important for detecting sarcasm, we perform the following feature importance analysis.

6.4 Feature importance analysis

As observed, different feature sets have different effects on the performance. While we may use many features to detect sarcasm, clearly, some features are more important than others. Therefore, we perform a thorough analysis of features to determine the features that contribute the most to detecting sarcasm. This analysis can be done with any feature selection algorithm. We use the odds-ratio (coefficients from ℓ_1-regularized logistic regression) for the importance analysis. The top 10 features in decreasing order of importance for sarcasm detection are the following:

1. Percentage of emoticons in the tweet.
2. Percentage of adjectives in the tweet.
3. Percentage of past words with sentiment score 3.
4. Number of polysyllables per word in the tweet.
5. Lexical density of the tweet.
6. Percentage of past words with sentiment score 2.
7. Percentage of past words with sentiment score -3.
8. Number of past sarcastic tweets posted.
9. Percentage of positive to negative sentiment transitions made by the user.
10. Percentage of capitalized hashtags in the tweet.

Interestingly, we observe that features derived from all forms of sarcasm: text expression-based features (1, 2, 5, 10), emotion-based features (3, 6, 7), familiarity based features (8), contrast-based features (9) and complexity-based features (4) rank high in terms of discriminative power.

6.5 Evaluating effectiveness of historical information

In our framework for detecting sarcastic tweets, we have included the user's historical information on Twitter in the form of past tweets. However, it might be computationally expensive to process and use all the past tweets for classification. Furthermore, it is unrealistic to assume access to so many past tweets for each user will be always available. Therefore, it is imperative that we identify the optimum number of past tweets to be used to detect sarcasm. To do this, we measure SCUBA's performance by executing the sarcasm classification multiple times while varying the number of past tweets available to us.

Figure 1 shows the performance obtained with varied past tweets (smoothened using a moving-average model). We observe that with no historical information, we obtain an accuracy of 79.38%, which still outperforms all baselines. Interestingly, using only the user's past 30 tweets, we obtain a considerable gain (+4.14%) in performance. However, as we add even more historical tweets, the performance does not significantly improve. Therefore, if computationally constrained, one can use only the past 30 tweets and expect a comparable performance.

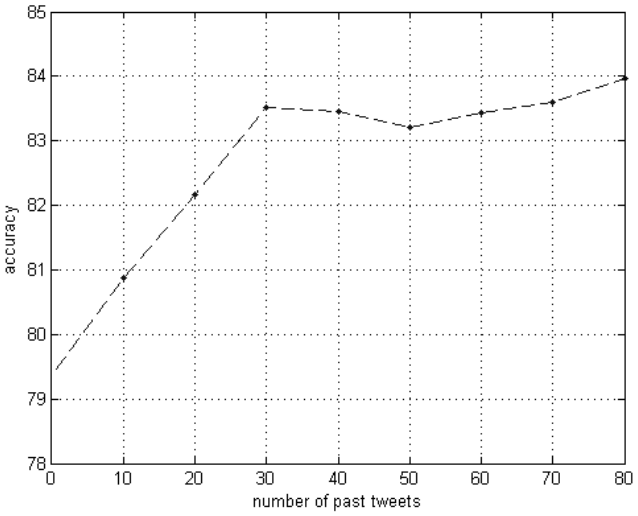

Figure 1: Effect of Historical Information on Sarcasm Detection Performance.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce SCUBA, a behavioral modeling framework for sarcasm detection. We discuss different forms that sarcasm can take, namely: (1) as a contrast of sentiments, (2) as a complex form of expression, (3) as a means of conveying emotion, (4) as a possible function of familiarity and (5) as a form of written expression. We construct relevant features to represent these forms on Twitter. We train a supervised learning algorithm using the constructed features to detect sarcastic tweets. Through multiple experiments, we demonstrate that SCUBA is effective in detecting sarcastic tweets. SCUBA's main two advantages are considering psychological and behavioral aspects of sarcasm and leveraging users' historical information to decide whether tweets are sarcastic or not.

Importantly, we have demonstrated that even limited historical information may greatly help improve the efficiency of sarcasm detection. This makes SCUBA a good fit for real-world, real-time applications which have high computational constraints. It is important to note that while we perform our evaluation and experiments on a Twitter dataset, SCUBA can be generalized to other social media sites. It can be easily expanded by including other site-specific features. This further widens the scope of applicability of SCUBA to different social media sites.

With nearly all major companies having a social media presence, SCUBA can complement existing sentiment analysis technologies to better serve the needs of consumer assistance teams online. With consumer assistance teams aiming for a zero-waiting time response to customer queries through social media, undetected sarcasm can result in embarrassing gaffes and potential PR disasters. Using SCUBA, social media teams can better detect sarcasm and deliver appropriate responses to sarcastic tweets.

In the future, we wish to expand SCUBA to also factor in users' social networks and their current and past interactions for sarcasm detection. This bodes well with existing research [31] which suggests that users are more likely to use sarcasm with friends than with strangers. Further, we wish to apply our behavioral modeling framework to detect other non-literal forms of language such as humor.

ACKNOWLEDGMENTS

This work was supported, in part, by the Office of Naval Research grants N000141410095 and N000141310835. We would also like to thank Dr. Heather Pon-Barry and Dr. Subbarao Kambhampati for their constructive criticisms and suggestions that helped improve this work.

8. REFERENCES

[1] M. Basavanna. *Dictionary of psychology*. Allied Publishers, 2000.

[2] F. Bi and J. A. Konstan. Customer service 2.0: Where social computing meets customer relations. *IEEE Computer*, 45(11):93–95, 2012.

[3] H. S. Cheang and M. D. Pell. Recognizing sarcasm without language: A cross-linguistic study of english and cantonese. *Pragmatics & Cognition*, 19(2), 2011.

[4] 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.

[5] M. L. Dress, R. J. Kreuz, K. E. Link, and G. M. Caucci. Regional variation in the use of sarcasm. *Journal of Language and Social Psychology*, 27(1):71–85, 2008.

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[7] T. Fawcett. An introduction to {ROC} analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. {ROC} Analysis in Pattern Recognition.

[8] R. Flesch. A new readability yardstick. *Journal of applied psychology*, 32(3):221, 1948.

[9] R. W. Gibbs. Irony in talk among friends. *Metaphor and symbol*, 15(1-2):5–27, 2000.

[10] R. González-Ibáñez, S. Muresan, and N. Wacholder. Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*, pages 581–586. Citeseer, 2011.

[11] H. P. Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and semantics*, volume 3. New York: Academic Press, 1975.

[12] H. P. Grice. Some further notes on logic and conversation. In P. Cole, editor, *Syntax and Semantics 9: Pragmatics*, pages 113–127. 1978.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[14] Y. Hu, K. Talamadupula, S. Kambhampati, et al. Dude, srsly?: The surprisingly formal nature of twitter's language. In *ICWSM*, 2013.

[15] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document, 1975.

[16] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538–541, 2011.

[17] R. J. Kreuz and G. M. Caucci. Lexical influences on the perception of sarcasm. In *Proceedings of the Workshop on computational approaches to Figurative Language*, pages 1–4. Association for Computational Linguistics, 2007.

[18] R. J. Kreuz and S. Glucksberg. How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of Experimental Psychology: General*, 118(4):374, 1989.

[19] S. Kumon-Nakamura, S. Glucksberg, and M. Brown. How about another piece of pie: The allusional pretense theory of discourse irony. *Journal of Experimental Psychology: General*, 124(1):3, 1995.

[20] C. Liebrecht, F. Kunneman, and A. van den Bosch. The perfect solution for detecting sarcasm in tweets# not. *WASSA 2013*, page 29, 2013.

[21] W. Liu and D. Ruths. What's in a name? using first names as features for gender inference in twitter. In *Analyzing Microtext: 2013 AAAI Spring Symposium*, 2013.

[22] G. H. McLaughlin. Smog grading: A new readability formula. *Journal of reading*, 12(8):639–646, 1969.

[23] J. Oller, John W. Scoring methods and difficulty levels for cloze tests of proficiency in english as a second language. *The Modern Language Journal*, 56(3):pp. 151–158, 1972.

[24] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390, 2013.

[25] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[27] J. W. Pennebaker, M. E. Francis, and R. J. Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, page 71, 2001.

[28] A. Reyes, P. Rosso, and D. Buscaldi. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12, 2012.

[29] E. Riloff, A. Qadir, P. Surve, and Silva. Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, pages 704–714. ACL, 2013.

[30] P. Rockwell. Lower, slower, louder: Vocal cues of sarcasm. *Journal of Psycholinguistic Research*, 29(5):483–495, 2000.

[31] P. Rockwell. Empathy and the expression and recognition of sarcasm by close relations or strangers. *Perceptual and motor skills*, 97(1):251–256, 2003.

[32] P. Rockwell. The effects of cognitive complexity and communication apprehension on the expression and recognition of sarcasm. *Hauppauge, NY: Nova Science Publishers*, 2007.

[33] P. Rockwell and E. M. Theriot. Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis. *Communication Research Reports*, 18(1):44 – 52, 2001.

[34] C. Strapparava and A. Valitutti. Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086, 2004.

[35] J. Tepperman, D. R. Traum, and S. Narayanan. " yeah right": sarcasm recognition for spoken dialogue systems. In *INTERSPEECH*, 2006.

[36] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[37] M. Toplak and A. N. Katz. On the uses of sarcastic irony. *Journal of Pragmatics*, 32(10):1467–1488, 2000.

[38] 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.

[39] A. Utsumi. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics*, 32(12):1777–1806.

[40] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth. Cursing in english on twitter. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 415–425. ACM, 2014.

[41] A. B. Warriner, V. Kuperman, and M. Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, pages 1–17, 2013.

[42] R. Zafarani and H. Liu. Connecting users across social media sites: a behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 41–49. ACM, 2013.