Sarcasm detection

Soumik Dey
Indiana University Bloomington
Bloomington, Indiana 47401
soudey@umail.iu.edu

Dhaval Niphade
Indiana University Bloomington
Bloomington, Indiana 47405
dniphade@iu.edu

Avinaba Dasgupta
Indiana University Bloomington
Bloomington, Indiana 47405
adasgupt@indiana.edu

ABSTRACT

Sarcasm is a linguistic tool that uses irony to mock or convey contempt. It is still one of the constructs of language that most humans do not completely comprehend. Sarcasm may not be necessarily ironic but may employ ambivalence and thus is heavily dependent on the context of speech or text. Detecting sarcasm is challenging due to a variety of factors that operate either or unison or individually to imbue sarcasm into the statement. With an increasing number of advances in natural language processing means that now allow us to predict the sentiment of a statement, our ability to detect sarcasm in written text is better than ever. Detecting sarcasm can help eliminate noisy data in training data inputs which are being used in other natural language processing applications. This paper provides a succinct summary and understanding of the approach we have adopted to successfully detect sarcasm in text. It also goes on to enlist the challenges encountered within the experiment as well as those that we may encounter while working on larger data sets.

CCS CONCEPTS

•**Computing methodologies** →**Rule learning;** *Classification and regression trees; Inductive logic learning;*

KEYWORDS

Classification, rule learning, boolean fucntion minimization, inductive logic programming, supervised learning, optimization

ACM Reference format:
Soumik Dey, Dhaval Niphade, and Avinaba Dasgupta. 2017. Sarcasm detection. In *Proceedings of ACM SIGKDD conference, Halifax, Nova Scotia - Canada, August 13–17,2017 (KDD'17),* 4 pages.
DOI:

1 INTRODUCTION

Sarcasm can be used in sentences of all sizes with variable grammatical structures and across a wide array of topics. Primarily, to understand sarcasm, one has to know the context of speech or text. Thus, to successfully detect sarcasm we must have premature knowledge of the subject or sarcasm which might either not be available or be alluded to within the text. In our experiments, we've mined through a large dataset of tweets collected directly from Twitter.Furthermore, we've mined Twitter data labelled with # sarcasm and arrived at the hypothesis that sarcastic tweets often have big contrast of sentiments. We've also added other features such as valence shifters and unigram modelling to yield results. Through our experiments we have only looked at linguistic features, however recent research has shown sarcasm to be a function of author salient features(Bomman and Smith 2015) [1].

KDD'17, Halifax, Nova Scotia - Canada

DOI:

2 PREVIOUS WORK

While sarcasm detection has been studied upon, it has not been experimented upon due to some of the challenges we run into that are illustrated in the latter part of this paper. On June 5, 2014 BBC reported that the U.S. secret service was looking for a software system that could detect sarcasm in social media data. In the past, Lunando and Purwarianti [8] presented their sarcasm detection classifiers, including a Nave Bayes and Support Vector Machine for analyzing Indonesian social media using features of negativity and a number of interjections. Similarly, Chun —Che Peng, Mohammad Lakis and Jan Wei Pan's [10] analysis of the problem yields a simple solution that employs a Nave Bayes classifier and a binary SVM to classify sarcastic data. However, both of them deliver poor results when contextual clues go missing within the training or testing data.

It is essential to understand the basis of contextual valence shifters and the research that has already been conducted on it. Texts on which research has been conducted includes "Contextual Valence Shifters" (Polyani et al. AAAI 2004) [12] and "Automatic Extraction of Contextual Valence Shifters" (Boubel et al. ACL 2013) [4] . Polyani studied and theorized the rules of contextual valence shifters from the language perspective. Boubel showed the computational part of it and the intricacies involving natural language processing. Kennedy and Inkpen [7] had also tried to analyze sentiment for Movie Reviews in their paper "Sentiment Classification of Movie Reviews Using Con —textual Valence Shifters" [13]. Sumanth and Inkpen had studied "How much does word sense disambigua —tion help in sentiment analysis of micropost data ? " where they had used the datset of SEMEVAL 2013 task and generated SentiWordnet scores for each text with the help of word —sense disambiguation. (Wil —son et al 2005) talks about the Phrase level sentiment analysis where valence shifters are used for Phrase level analysis of Sentiment.

3 DATA

Prior work on sarcasm detection on Twitter (Gonzalez et. al. 2011) found low agreement rates between human annotators at the task of judging the sarcasm of others'tweets; consequently, recent research exploits users'self —declarations of sarcasm in the form of # sarcasm or # sarcastic tags of their own tweets. This design choice does not capture the likely more common varieties of sarcasm expressed without an explicit hashtag, but does yield positive examples with

high precision. We follow the same method and collect 2000 tweets with # sarcasm and 2000 normal tweets.

3.1 Preprocessing

There is a drawback to taking our data from Twitter; it's noisy. Some people use the # sarcasm hashtag to point out that their tweet was meant to be sarcastic, but a Human would not have been able to guess that the tweet is sarcastic without the label # sarcasm (example: What a great summer vacation I've been having so far :) # sarcasm). One may argue however that this is not really noise since the tweet is still sarcastic, at least according to the tweet's owner, and that sarcasm is in the eyes of the beholder. The converse also happens, someone may write a tweet which is clearly sarcastic but without the label # sarcasm. There are also instances of sarcastic tweets where the sarcasm is in a linked picture or article. Sometimes tweets are responses to other tweets, in which case the sarcasm can only be understood within the context of the previous tweets. Sometimes the label # sarcasm is meant to indicate that, while the tweet itself is not sarcastic, some of its hashtags are (example: Time to do my homework # yay # sarcasm).

Before extracting features from our text data it is important to clean it up. To remove the possibility of having sarcastic tweets in which the sarcasm is either in an attached link or in response to another tweet, we simply discard all tweets that have http addresses in them and all tweets that start with the @ symbol. Ideally we would only collect tweets that are written in English.

4 FEATURES

We have performed feature extraction from the data, namely unigram modelling, sentiment extraction and topic modelling using Latent Dirichlet Allocation with the help of scikit —learn library and SentiWordnet Lexicon

4.1 Unigrams

The first set of features consisted of tf —idf scores of the unigrams present in each dataset. We will refer to these features as the Unigram features. We performed experiments using bigrams and trigrams as well, but we found that including these features yielded little or no improvement over using unigrams alone, so we do not report results for these features here.

4.2 Sentiment

Using the part of speech tagger available in the Natural Language Tool Kit (NLTK) we've annotated the data with its respective part of speech. The data is then processed through a custom written valence shifter program that detects whether the given text has any cues for shifting the overall sentiment score. As its output it generates a final sentiment score that is based on the previous step. For computing the sentiment score through the lexical valence shifter tool that we've built, we explore a number of contextual valence shifters such as:

- Negatives and intensifiers
- Modals
- Presuppositional items
- Connectors

Some of the in —explorable shifters include:

- Cultural constraints
- Genre and Attitude Assessment
- Multi —entity evaluation
- Subtopics

Lexical valence in texts is the elocutionary force through the choice of synonyms chosen to depict the persons, events and situation involved. Observations such as these have led researchers to classify terms as positive or negative. The simple computation of the attitude expressed in a text would then consist of counting the negative and positive instances and decide on the basis of the highest number. We worked on mainly Sentence Level Valence Shift. Sentence Level Valence Sift is of two kinds. Negation handling —and modifiers namely enhancers and diminishers. Negation handling has been done in this case with a window size of one. That is negative words reveres the polarity of the next word in context within the window size of 2.

Example 4.1. *He is not good. Good* $= +1$
$Notgood = -1$

Modifiers and diminishers work in the same way except tone is scaled up or down.What I have done is look out for adverbs or adjectives using the pos_tagger of NLTK which uses the Penn Treebank and scaled up or down the tone of the next word with a window size of 1. The scaling up or down of the diminisher of modifier by using it as a divisor quotient for the sentiment score of the next word.

Example 4.2. *He is barely good.*
Good $= +1$
Barely $= 1/2$
Barely good $= +0.5$

Example 4.3. *He is extremely good.*
Good$= +1$
Extremely$= 2/3$
Extremely Good $= +1 * 3/2 = +1.5$

Sometimes the contributions made by various lexical items combine in a way that cannot be accounted for. For example, consider the sentence "The very brilliant organizer failed to solve the problem". This sentence has two essential parts to evaluate; one being the brilliance of the organizer that is augmented by the valence shifter "very" and the second one being his inability to solve the problem. The phrase "failed to solve the problem" reverses the sentiment of the sentence to produce a negative score. This can be illustrated as follows:

Example 4.4. *Brilliant +2 Original valence is adjusted by very*
Very brilliant +3 —3 Adjusted because of fail
Failed —1 —1
Solve the problem +1 0 Neutralized by fail

Analogously, connectors such as although, however, but, on the contrary, notwithstanding etc. can introduce information and act n information elsewhere in the text. For example, consider the sentence

Although Chris is excellent at sports, he is a horrible basketball player

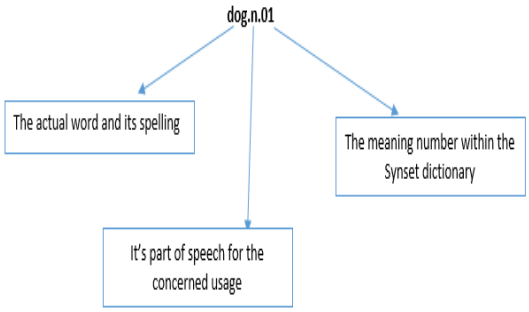

Figure 1: Synsets explained

While the statement assesses Chris'skills in sports to be excellent and thus affixes a positive sentiment/valence to the statement, the although combined with the negative assessment in the sentence's main clause he is a horrible basketball player effectively negates the positive force of the evaluation as applied to Chris. Thus, the base valence of terms could be imagined as:

Example 4.5. *Excellent +2*
Horrible —2
Total Score 0

Example 4.6. *Adjusted computation*
(Although) brilliant 0
Horrible —2
Total Score 0

In order to compute these individual sentiment scores, we leveraged the Senti —Wordnet [5] lexica in harmony with NLTK in Python (version 2.7) [2], where we calculated sentiment scores of each word with valid sentiment associated with them and tagged them with three types of score —positive, negative and neutral (also referred to as objective). Subsequently, we've averaged these scores over the total number of words in an individual tuple of the dataset. This has been done taking into account the sense of the word, using word sense disambiguation with the help of NLTK.

4.3 Synsets

Finally, the third set of features are the sense —imbued unigram features where each word has been assigned additional sense and represented in synset format. This helps in identifying the word sense usage within the given context —a process that is often called synset formatting. The word senses are represented by 3 attributes as shown in the diagram above.

The NLTK wsd package is not very famous for accurate sense disambiguation. Therefore sentiment scores are not that accurate however the process is dynamic rather than static access of a lexicon like general lexicon. The synsetting however is much more exact because when wsd makes errors it is consistent and hence the sparse matrix becomes consistent.

Table 1: Results

Algorithm	Unigram	Synset	Topics	Sentiment	All
Naive Bayes	71.07%	61.98%	61.98%	60.60%	71.074%
Logistic Regression	77.68%	71.90%	64.73%	59.77%	77.96%
SVM	75.48%	70.79%	62.80%	60.05%	77.68%

4.4 Topics

There are words that are often grouped together in the same tweets (example: saturday, party, night, friends, etc.). We call these groups of words topics. If we first learn the topics, then the classifier will just have to learn which topics are more associated with sarcasm and that will make the supervised learning easier and more accurate. To learn the topics, we used the python library scikit —learn [9] which implements topic modeling using latent Dirichlet allocation (LDA). We first feed all the tweets to the topic modeler which learns the topics. Then each tweet can be decomposed as a sum of topics, which we use as features.

5 EXPERIMENTS AND RESULTS

The goal of our experiments was to discover which set of features and which classifiers would yield the greatest performance on each personality factor as measured by the F —1 score and accuracy. We experimented with each set of features described in the previous section individually and all together and while I experimented with many types of classifiers, we only report the results here for the Gaussian nave Bayes classifier (NB), logistic regression (LG), and linear support vector machines (SVM). All experiments were conducted using the scikit —learn python package (Pedregosa, 2011).

Grid search was conducted for the logistic regression and linear support vector machine classifiers to find the optimal regularization parameter, and the value of the parameter that yielded the highest 10 —fold cross —validation accuracy was used for further testing (each fold was stratified so that each fold would have approximately the same class proportions).

The baseline was a simple classifier that always responded with the majority class.We combined normal unigrams and sentiment and synset and unigrams to get combined classifiers as well. The performance of logistic regression and support vector machines was similar in most cases, but Logistic Regression was more often the best classifier. The All classifier has been restricted to unigrams, Sentiment and Topics as it gives the best result.

6 CHALLENGES

In addition to some of the valence shifters described in the earlier sections, a number of contextually dependent valence shifters need to be accounted for, such as:

- Modals —Modal operators setup a context of possibility or necessity and in texts they initiate a context in which valenced terms express an attitude towards entities which do not necessarily reflect the author's attitude towards those entities in an actual situation under discussion. Assume the sentence: "Mary is a terrible person". She is mean to

her dogs. Here terrible and mean are negatively valenced terms. Thus the score for each of the sentences would have be −1. However, if the sentence would have been "If Mary is a terrible person, she ought to be mean to her dogs". Here the modal operators neutralize the overall sentiment of the sentence and thus the score should be zero.

- Presuppositional items —Here some of the preceding or succeeding terms shift the valence of evaluative terms. This is typical for adverbs such as "barely" which shift the sentiment score of the succeeding word. For example, in the phrase barely sufficient, "barely" shifts the valence of the positive term sufficient to indicate that better was expected. Thus, if the sentiment score of sufficient were +2, through the use of a presuppositional item it is reduced to +1.
- Multi —entity evaluations —Most documents are composed of a wide variety of entities. For example, in a product review the reviewer may write extensively about one flaw but may praise a number of the rest of the features in fewer lines. In this case it would be incorrect to assume that the overall sentiment of the review was negative. Thus, a simple comparison of number of positive terms versus number of negative terms would yield incorrect results.
- Genre and Attitude Assessment —The assessment of the author's attitude may be greatly influenced by the genre of communication in which the valence marked terms occur. Assessing attitude in a document in which there are various participants "speaking" or "narrating" can be a challenge.
- Reported Speech —Consider the sentence "Mary is a slob". This sentence would be evaluated to a score of —1. Now consider "John said Mary was a slob" —here the author asserts that John believes Mary is a slob and he may or may not agree with John thus making the score of the sentence neutral. Now consider, "John said Mary was a slob and he is right" —here the negative influencer slob would be counted along with the positive influencer right. Thus, Reported Speech and Thought Operators such as these need not necessarily cancel out each other and the overall sentiment of the statement could be either of the three possibilities.
- Author Salient Features —While features derived from the author yield the greatest improvements in accuracy over the tweet alone, all feature classes (response, audience and author) display statistically significant improvements over the tweet-only features that ignore the communicative context. This confirms an effect on the interaction of the author and audience in the recognition of sarcasm, which can lead us to ask: who is this audience, and what about them is predictive of sarcasm across users? But we did nt have time or resources toimplement this.

7 CONCLUSION

As evident from the examples provided in this paper as well from our experiments, detecting sarcasm is a challenging task. If we need to build a classifier that detects sarcasm in all topics, we need to sample much larger data. Obtaining new features that are relevant to sarcasm is an essential next step in sarcasm detection.

Through our experiments we deduced that Nave Bayes, One class SVM yielded poorer results than Logistic Regression. We've also observed that accuracy depends on a combination of the feature types. Unigrams and Bigrams are alone insufficient in designing an accurate classifier. When combined with other types such as topic modelling and Synset, the accuracy is expected to greatly increase since it provides context. This is in comparison to some other results if not better than what was previously achieved (Smith 2015).

REFERENCES

[1] David Bamman and Noah A. Smith. 2015. Contextualized Sarcasm Detection on Twitter. In *ICWSM*.

[2] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python* (1st ed.). O'Reilly Media, Inc.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022. http://dl.acm.org/citation.cfm?id=944919.944937

[4] Nomi Boubel, Thomas Franois, and Hubert Naets. 2013. Automatic extraction of contextual valence shifters.. In *RANLP*, Galia Angelova, Kalina Bontcheva, and Ruslan Mitkov (Eds.). RANLP 2013 Organising Committee / ACL, 98–104. http://dblp.uni-trier.de/db/conf/ranlp/ranlp2013.html#BoubelFN13

[5] Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LRECfi06.* 417–422.

[6] Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in Twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2 (HLT '11).* Association for Computational Linguistics, Stroudsburg, PA, USA, 581–586. http://portal.acm.org/citation.cfm?id=2002736.2002850

[7] Alistair Kennedy and Diana Inkpen. 2006. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence* 22 (2006), 2006.

[8] Edwin Lunando and Ayu Purwarianti. 2013. Indonesian social media sentiment analysis with sarcasm detection. In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on.* IEEE, 195–198.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[10] Chun-Che Peng, Mohammad Lakis, and Jan Wei Pan. 2015. Detecting Sarcasm in Text: An Obvious Solution to a Trivial Problem.

[11] Livia Polanyi and Annie Zaenen. 2004. Contextual Valence Shifters. In *Working Notes — Exploring Attitude and Affect in Text: Theories and Applications (AAAI Spring Symposium Series).*

[12] Livia Polanyi and Annie Zaenen. 2005. Contextual valence shifters. In *Computing Attitude and Affect in Text.* Springer. MISSING

[13] Chiraag Sumanth and Diana Inkpen. 2015. How much does word sense disambiguation help in sentiment analysis of micropost data?. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.* Association for Computational Linguistics, Lisboa, Portugal, 115–121. http://aclweb.org/anthology/W15-2916