Sarcasm Detection in Twitter

Ameesha Mittal, Divakar Verma, Abhishek Joshi, Shyamal Vaderia, Nischay Singh, Darshil Kapadia, Vatsal Gupta

Birla Institute of Technology and Science, Pilani

***Abstract*— Sarcasm transforms the polarity of an apparently positive or negative utterance into its opposite. While a fair amount of work has been done on automatically detecting emotion in human speech, there has been little research on sarcasm detection. Current approaches to automatic sarcasm detection rely primarily on lexical and linguistic cues. In this work, we have followed a research paper (mentioned in I), implemented a part of that paper with our own improvisation and compared and contrasted our results with theirs. We have applied behavioral approach to sarcasm detection on twitter dataset.**

***Index Terms*—Sarcasm detection, twitter, behavioral modelling, ensemble classification, SCUBA.**

I. Introduction

Sarcasm Detection is a particularly difficult task, even for humans. Sarcasm is an anti-thetic form of expression in which the meaning implied is opposite to literal message conveyed. Thus, the intentional ambiguity involved makes the aforementioned task an especially challenging one.

Automatic detection of sarcasm is still in its infancy. One reason for the lack of computational models has been the absence of accurately-labeled naturally occurring utterances that can be used to train machine learning systems. [1]. In Twitter, messages can be annotated with hashtags such as #bicycling, #happy and #sarcasm.These hashtags have been used to build a labeled corpus of naturally occurring sarcastic, positive and negative tweets. We utilized this corpus for our project.

A. Background

With the increasing relevance of social media platforms in all fields, ranging from politics to customer service, almost all major MNCs have specialized teams dedicated to handling their social media outreach. Many customers prefer to send their feedbacks, comments, and opinions via these platforms, especially Twitter, because of the instantaneity provided. With the increasing customer expectations, minor goof-ups on the part of the customer response team results in tarnishing the reputation of the company. Hence there is a need of a robust sarcasm detection system.

Starting with the earliest known work by [2] which deals with sarcasm detection in speech, the area has seen wide interest from the natural language processing community as well. Following that, sarcasm detection from text has extended to different data forms (tweets, reviews, TV series dialogues), and spanned several approaches (rule-based, supervised, semi-supervised). This synergy has resulted in interesting innovations for automatic sarcasm detection.

This paper has been submitted to Dr. Yashvardhan Sharma, Dept. of Computer Science BITS Pilani, India, In the partial fulfillment of the requirement of the course CS-F415: Data Mining.

B. Motivation

The major motivation behind choosing this paper was that not only is the the problem tackled by it extremely interesting, but it also has seemingly endless real-world applications. Moreover the methodology adopted and the features created are ingenious and intuitive at the same time. The systematic approach followed in modelling something as complex as the psychological behaviour and cognitive ability of the user, using objective features, gives us an insight into how data mining transcends the boundaries of conventional statistical analysis to have applications a vast variety of disciplines.

C. Objective

Primarily, our project aims at analysing, implementing, remodelling and verifying the results published in [3]. The problem statement has been reproduced here for the sake of clarity:

"Given an unlabeled tweet t from user U along with a set of U's past tweets T, a solution to sarcasm detection aims to automatically detect if it is sarcastic or not."

Through this objective we aimed at gaining practical knowledge of different data mining techniques and a general understanding of the problem solving approach for different data mining tasks. We further aimed at gaining some insights to different classification techniques, feature formation etc.

The approach we used for sarcasm detection could be extended to identify humor, satire and irony and other complicated literary devices.

II. Related work

Sarcasm Detection is a topic of deep study in the field of Psychology, and there are extensive definitions and notions for the same. We draw similar ideas from this field to model our behavioral approach towards the problem.

There is a great deal of past and on going work related to sarcasm detection [4] [5]. Tepperman et al. conducted experiments using prosodic, spectral, and contextual cues to automatically identify sarcasm in the phrase 'yeah, right' [2]("'Yeah Right': Sarcasm Recognition for Spoken Dialogue Systems").

While research on automation of sarcasm detection in speech utilizes prosodic, spectral and contextual features, variation of pitch, loudness and treble are used among others. However, sarcasm detection in text has always relied on identifying text patterns [6] and lexical features. For example, many of the

existing approaches simply use the Bag of Words model with no relation with sentiments of the user.Davidov et al. [6] have devised a semi-supervised technique to detect sarcasm in Amazon product reviews and tweets whereby 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.

Hence it is of utmost importance to employ theories from behavioral and psychological studies to construct a behavioral modeling framework tuned for detecting sarcasm.

III. Proposed technique(s) and alogorithm(s)

As suggested by the author, sarcasm can be expressed in 5 forms:

1. Sarcasm as a constraint of sentiments.
2. Sarcasm as a complex form of expression
3. Sarcasm as a means of conveying emotion.
4. Sarcasm as a possible function familiarity.
5. Sarcasm as a form of written form of expression.

Among these, we are implementing the first, second and the fifth approaches, as assigned to us. The descriptions of these forms have been extensively covered in the initial presentation and hence have not being covered here, so as to avoid unnecessarily lengthening of the report. Instead, the report stresses upon the details of our implementation, improvisations and the results obtained.

In general, we will be using 12 different classification algorithms on the feature sets generated and will be comparing their results. These have been mentioned below:

1. Gaussian Naive Bayes
2. Stochastic Gradient Descent
3. Decision Tree
4. Random Forest
5. Adaboost
6. L1 regularised Logistic Regression
7. L2 regularised Logistic Regression
8. Gradient Boosting
9. Bagging Model
10. SVM (rbf kernel)
11. K-neighbours hybrid
12. SVM (polynomial kernel)

We also aim to review the author's assertion that the following features contribute the most the classification process.

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. Percentage of positive to negative sentiment transitions made by the user.
9. Percentage of capitalized hashtags in the tweet.

IV. our Improvizations

We did several improvizations at each stage of the project:

A. *Pre-processing*

We used lemmatization during pre-processing to reach to the base of any word in the tweet. Lemmatization is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech.

B. *Feature Generation*

1) Dealing with joined words: It is a common trend to use hashtags in tweets. These hashtags contains several words clubed together. Thus while extracting sentiment score directly for complete words in a tweet we miss on the words contained in these hashtags. Therefore, we instead consider each and every subset of every word in the tweet for sentiment extraction.

2) Role of emoticons: Emoticons have become a popular means of showing emotions. Thus they provide a good deal of information regarding a person's mood, sentiments etc. We have utilized emoticons to get two extra features in the first approach where we view sarcasm as a contrast of sentiments. A set of 50 emoticons have been labelled with suitable sentiment value and used to extract total positive and total negative sentiments as expressed through emoticons.

3) POS tagging[7]: There are 35 different POS tags provided by nltk library of python. The author;s approach is to find POS tags of capitalized words in the tweets and find the probability of those tags. However if we extract the probability for each of these tags, we will end up with a highly sparse feature set. Moreover some of the tags are not much relevant to our problem. Thus we extract 10 sets of relevant POS tags where some sets are a union of several POS tags eg all kinds of verbs have been combined under a single tag.

C. *Classification*

1) We used Ensemble Classifiers along with others, over the generated feature set for various train-test splits to find the optimum value. Some of the tried classifiers include RandomForest, Adaboost, Gradient Boosting, etc.

2) Owing to the limited computational power available to us, we used random sampling for some of the costly algorithms, namely SVM classifier with polynomial kernel.

3) We used a hybrid classifier in which the results of the K-neighbours classifier were fed to bagging model classifier to improve the performance by a substantial margin.

V. Dataset used

The initial dataset provided by the author consisted of Tweet IDs (unique IDs assigned to tweets by twitter) and their respective classification tags i.e. Sarcastic or Non-Sarcastic.

The dataset was further elaborated using Tweepy- a python library built as a wrapper on the Twitter API easy use of the same in python.

Tweet details- namely text, date-time and the author name were extracted from twitter servers using Tweepy.

Another dataset was used from [8] which consists of 16 lacs tweets with a tag for 'positive', 'negative' or 'neutral'

VI. Experiments and results

Application of the techniques proposed by the author along with our improvisations yielded some very intriguing results.

A. Methodology

We divided our features into four feature sets as suggested by the author namely :

1. Sarcasm as a contrast of Sentiments (Set 1)
2. Sarcasm as a complex form of expression (Set 2)
3. Sarcasm as a written form of expression (Set 3)
4. All of these features combines (read, BigFeautreSet)

Various algorithms were applied on these feature sets to measure which feature set contributed towards the prediction to what extent.

We also employed the idea of varying training-testing split on the dataset to get a vague idea of the location of optimum results.

B. Evaluation and Results

1. Set1 accuracy table:

Classification Algorithms	**Split Ratio**		
	50: 50	**25: 75**	**10: 90**
Gaussian Naive Bayes	56.60%	57.34%	59.21%
SVM (rbf kernel)	54.39%	54.39%	54.39%
Decision Tree	71.91%	72.53%	71.12%
SVM (polynomial kernel)	66.20%	64.95%	67.28%
Random Forest	71.45%	70.36%	72.08%
Adaboost	77.11%	77.11%	77.11%
L1 regularised Logistic Regression	75.73%	75.73%	75.73%
L2 regularised Logistic Regression	75.60%	75.60%	75.60%
Gradient Boosting	79.12%	79.09%	79.09%
Bagging Model	75.17%	75.07%	74.78%

2. Set 2 Accuracy table:

Classification Algorithms	**Split Distribution**		
	50: 50	**25: 75**	**10: 90**
Gaussian Naive Bayes	57.39%	57.80%	57.73%
Stochastic Gradient Descent	53.86%	54.90%	52.96%
Decision Tree	55.44%	54.50%	53.13%
SVM (polynomial kernel)	66.20%	64.95%	67.28%
Random Forest	56.73%	58.52%	54.61%
Adaboost	61.04%	61.22%	63.48%
L1 regularised Logistic Regression	60.35%	60.76%	60.69%
L2 regularised Logistic Regression	60.22%	60.89%	60.8%
Gradient Boosting	61.01%	61.35%	65.95%
Bagging Model	56.50%	55.10%	55.09%

3. Set 3 Accuracy table

Classification Algorithms	**Split Distribution**		
	50: 50	**25: 75**	**10: 90**
Gaussian Naive Bayes	70.07%	74.25%	75.00%
Decision Tree	78.17%	77.09%	74.34%
SVM (rbf kernel)	82.25%	83.08%	84.04%
Random Forest	82.84%	82.88%	84.21%
Adaboost	83.93%	84.33%	83.38%
L1 regularised Logistic Regression	81.98%	82.48%	81.25%

L2 regularised Logistic Regression	81.85%	82.35%	80.92%
Gradient Boosting	84.82%	85.12%	85.19%
Bagging Model	83.17%	83.54%	83.88%
Extra Trees	81.26%	81.56%	82.56%
Kneighbours	75.96%	76.95%	76.97%
KneighboursHybrid	80.73%	81.83%	66.11%

4. Total Feature Set Accuracy table

Classification Algorithms	**Split Distribution**		
	50: 50	**25: 75**	**10: 90**
Gaussian Naive Bayes	71.12%	72.08%	71.38%
SVM (rbf kernel)	56.93%	55.76%	57.56%
Decision Trees	77.54%	77.22%	77.13%
Random Forest	83.89%	83.93%	84.04%
Adaboost	84.49%	85.38%	84.04%
Bagging Model	84.42%	84.85%	84.86%
Extra Trees	80.80%	81.23%	83.88%
Gradient Boosting	85.14%	85.71%	85.03%
KneighboursHybrid	61.63%	66.16%	67.11%
L1 regularised Logistic Regression	82.81%	82.75%	80.59%
L2 regularised Logistic Regression	83.00%	82.88%	80.92
Kneighbours	57.19%	55.76%	55.92%

C. *Error Analysis*

Set 1 Analysis: The performance of all algorithms is more or less stable, the algorithms when compared to one another give varied results. While some algorithms like SVM (rbf kernel) perform as poor as 54%, some robust algorithms like Gradient Descent Ensemble perform extremely well at an accuracy of about 79%. Hence we can safely conclude that Set 1 is moderate predictor. Also it must noted that this dataset is pathological to multi-layered perceptron model and Stochastic Gradient Descent

Set 2 Analysis: All the 11classification algorithms that we applied performed poorly on the features of Set Hence, we may infer that Set 2 as a whole is a poor predictor. The best performance is obtained by using the Gradient Boosting, at an accuracy of 65.95%, while the worst performance was obtained by applying the Stochastic Gradient Descent at 52%.The results of SVM polynomial kernel algorithm have not been displayed as they are highly inconsistent on repeated runs.

Set 3 Analysis: This feature set is by far the best predictor giving an accuracy of as high as 85%. Even relatively primitive algorithms such as Gaussian Naive Bayes give a pretty satisfactory result of about 75%.

Full Feature set Analysis: By running classification algorithms on different features sets individually, and then running them on the BigFeatureSet, we drew certain conclusions. Gradient boosting gives the best accuracy of 85.71% on a train-test split of 75:25. K-neighbours and SVM (rbf kernel) give a below-par accuracy of about 55%. All the other ensemble classifiers give a satisfactory performance.

General Observations: In general, ensemble methods always outperform singleton classifier algorithms, showcasing consistency with the theoretical expectation. Gradient Boosting works the best, for all feature sets individually, as well as on the BigFeatureSet. Multi-Layered Perceptron model gives the most absurd, inaccurate and inconsistent results. In general, the 90:10 train-test split gives the best accuracy results. In coherence with the author's assertion, L1 and L2 regularized logistic regression, gives a result almost at par with Gradient Boosting. The comparatively better performance of primitive algorithms on Feature Set 3 as compared to other feature sets is probably because of the sparsity of the data produced by the feature set 3.

VII. Conclusion and Future Work

As is evident from the results described above, even the partial implementation (only features of sections 5.1, 5.2, 5.5 have been implemented) of the SCUBA framework gives pretty robust results. This clearly indicates that incorporating features that describe the psychological and behavioral aspects of the user goes a long way in helping the process of automatic identification of sarcasm.

Because of the encouraging results obtained, future work could definitely be pursued in the direction of expanding the feature set in order to include more features that are expressive of the user's behaviour. Moreover, the detection of sarcasm in tweets is limited, in the sense that the sentences are of limited sizes (Twitter has a limit of 140 characters per tweet). The same approach could be expanded to detect sarcasm on other social media platforms (To classify posts as sarcastic on facebook for example). This could go a long way in restricting

the spread of fake news on account of people not recognizing the posts as sarcastic.

Given more time and more computational power, a greater range of algorithms could be applied, more features could be generated and processed, and visualization too could be employed in order to get a better grasp of the results obtained. Finally, as the approach of incorporating behavioral analysis successfully improves sarcasm detection, the same could be employed in order to identify equally (or perhaps even more) complex forms of expression such as humor, satire, etc.

Moreover, the dataset of sarcastic tweets has been created by extracting tweets having #sarcasm and #not tags. The expectation of the users, while writing this tweet, might have been that without these hashtags, identifying the tweets as sarcastic might be very difficult. Hence, there is a strong possibility that our dataset of sarcastic tweets may be biased towards the hardest forms of sarcasm. Given more time, a better approach to building an unbiased dataset of sarcastic tweets can be developed.

On an ending note, the current approach works on a static dataset. The possibility of adding incremental classification capabilities could also be attempted in the future.

VIII. WORK UPDATE

A. *Work Completed*

Among the 335 features proposed in the research paper, to model 5 different forms of sarcasm, a total of 58 features, completely modelling 3 different forms of sarcasm, namely section 5.1.1 (Sarcasm as contrast of sentiments), 5.1.2 (Sarcasm as a complex form of expression) and 5.1.5 (Sarcasm as a written form of expression) have been implemented. (We were enthusiastic to create the rest as well, but they weren't assigned to the group).

12 different classifier algorithms, each with 3 different train-test splits, were run on the set of 58 features, the results of which have been elaborately described in section 5.

B. *Work Remaining*

The features described under the three remaining subheadings (5.1.3 and 5.1.4) are yet to be developed for the full implementation of the paper. We have also not yet developed an interface to accept new tweet and classify it.

IX. REFERENCES

[1] González-Ibánez, Roberto, Smaranda Muresan, and Nina Wacholder. "Identifying sarcasm in Twitter: a closer look." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*.

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

[3] Rajadesingan, Ashwin, Reza Zafarani, and Huan Liu. "Sarcasm detection on twitter: A behavioral modeling approach." *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 2015.

[4] Rakov, Rachel, and Andrew Rosenberg. "" sure, i did the right thing": a system for sarcasm detection in speech." *INTERSPEECH*. 2013/

[5] Maynard, Diana, and Mark A. Greenwood. "Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis." *LREC*. 2014.

[6] 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

[7] Derczynski, Leon, et al. "Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data." *RANLP*. 2013.

[8] Naji, Ibrahim. "Twitter Sentiment Analysis Training Corpus (Dataset)" . Web. 22 Sep. 2012. Link: http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/