Sarcasm Detection in Article Headlines

Christopher Pillay, 1362077
The University of the Witswatersrand
Johannesburg, South Africa
1362077@students.wits.ac.za

***Abstract*—The topic that this paper will be based on is sarcasm. Sarcasm can be viewed as a complex form of irony, its use is seen throughout the internet and features in most types of media. The aim of sarcasm is to implicitly convey information within a message to an audience or another person. In this paper we aim to address the issue of detecting sarcasm on article headlines to provide a basic understanding to reader about the article's writing style. The dataset consists of headlines from different websites and whether they are sarcastic or not, once the data is pre-processed using natural language processing techniques, a machine learning algorithm such logistic regression is utilized in the model which is then fitted with the data. After training and testing the highest accuracy our approach obtains is 82.18%.**

I. Introduction

Over the years the number of people who make use of the internet for news, entertainment and social media has increased drastically. With the increased number of online users comes an increase in online businesses, news vendors, blogs, campaigns and more. Therefore there is no shortage of articles for users to browse through and content creators such as news vendors and bloggers have to compete to get users to sign up as subscribers, for the purpose of gaining recognition and earning of income through advertisement campaigns on their websites.

There are many methods that employed in persuading or leading an online user to their articles. One of the most important and widely used methods is through captivating or exaggerated headlines to get the reader's attention and create interest. Sarcasm is one method that is commonly utilized to pique the interest of a reader. It is defined as *"the use of irony to make or convey contempt"*, this is the formal definition of sarcasm which is found on the online Oxford dictionary. People may view sarcasm differently throughout the world, some may approve of it while others may not be as fond of it as it may not be obvious, either way it is highly used in the media and people use of it in their daily lives in subtle and overly obvious usage [1].

A. Motivations

Sarcasm identification may aid in the process of sentiment analysis when it is applied to article headlines on various websites. The appearance of headlines may be misleading. "Thank you Eskom for being so reliable!!!", may be considered a headline that compliments the electricity generation company, especially to readers who do not know the company whereas it is clearly evident as being sarcastic when viewed by a citizen of South Africa. All readers have different preferences due to personalities or other factors, therefore while some readers may enjoy the sarcastic nature of some articles others may not which leads to irritation as they may not like sarcasm or fail to grasp its intention [2]. Therefore this paper looks at differentiating articles by their headlines so that a reader can decide to read an article or not based on whether they approve the use of sarcasm or not.

B. General explanation of Proposed approach

In this paper a dataset which consists of article headlines from different websites will be used as input, the data also specifies whether or not the article is sarcastic.The headlines will undergo pre-processing using natural language processing techniques such as tokenization, word stemming and normalization for tense variations. For our machine learning algorithm the sigmoid function will be utilized in our prediction function and batch gradient descent will be used as the learning function. We will fit the models multiple times to determine which parameters provided suitable error rates. The parameters of the model with the lowest error rates will then be utilized for the final model which will be fitted and used for prediction to provide multiple statistics to analyze the model performance such as accuracy, recall, precision and the f1 score. A cross validation version of the model with the chosen parameters will also be fitted to get a better accuracy and error rate, and a built-in SVM model will be fitted with data to act as a baseline and a comparison for our initial logistic regression model.

II. Related work

Over the last 15 years there has been more interest shown towards sentiment analysis and towards online users, as researchers and companies want to learn more about user/customers and their experiences to be able to understand what will effectively persuade or influence them later on based on information that is left online. This can be in the form of reviews, comments or social media postings. Across research papers the classification methods and features utilized may differ due to varying aims of the research. Sarcasm has been looked at from a neurobiological [3] and psychological

perspective [4] but it has also been studied and viewed as a linguistic behavior which plays a role in characterization of a person. Researchers have therefore attempted to identify and detect sarcasm automatically in statements, in the context of being a linguistic behaviour [2]. Even though there have been studies that claimed that sarcasm is not a linguistic phenomenon, there has been a large amount of research on the topic that have achieved impressive results in terms of accuracy and precision.

If we look at the work of Burfoot and Baldwin [5], satirical articles were filtered from newswire documents where a unique set of features was utilized which included slang and profanity, SVM's were implemented to perform classification of articles that contained satire. There has been a lot of papers that focus on the task of sarcasm detection, the work of Tepperman et al. [6] analyzes certain expressions and how often they occur and whether they appeared in a context that could be classified as sarcastic or not. The approach is effective and accurate when deciding whether a specific expression or phrase is sarcastic but fails to generalize for other possible types of sarcasm that exist. The work of Barbieri et al. [7] took an approach of classifying text into four different groups, irony, sarcasm, politics and humour.

A method which is featured in Riloff et al. [8] enables the detection of a specific type of sarcasm in which a sentiment that is positive will be contrasted with a negative situation. The method employed a bootstrapping algorithm which utilized a seed word and learnt text which showed positive sentiment with phrases that cited negative situations. Although the method had potential on the dataset, most of the sarcastic text does not fit into the category the paper focuses on, furthermore the approach is reliant on *"negative situations"* which are present in the training set and is unable to handle new text.

The work of Rajadesingan et al. [2] takes a more in-depth approach which looks at the psychology behind sarcasm, by employing a behavioural model for sarcasm detection on Twitter. Many forms of sarcasm were able to be successfully identified due to the use of older tweets of users. Extraction of features for this method depends on the previous data collected from a user, therefore if a person makes use of sarcasm regularly their tweets will have a higher chance of being classified as sarcastic. A possible issue is that the above method would not perform in a real-time sarcasm detection setting. Most models mentioned above achieve adequate results while some perform better than others. For in-depth analysis of results one can refer to the articles.

III. Dataset and features

The name of the dataset utilized in this paper is News Headlines Dataset For Sarcasm Detection which was uploaded by Rishabh Misra [9] and can be found on the kaggle website and is in the json format. The dataset was collected from two websites to overcome limitations of Twitter datasets which usually consist of data which tend to have a lot of noise due to labels and language used. *TheOnion* news website specialises in sarcastic headlines for their articles while the *HuffPost* news website publishes articles with normal headlines. The headlineds collected from both website are of quality as they are professionally written, therefore there is less language errors or spelling mistakes. This helps with the reduction of sparsity and pre-trained embeddings have a higher chance of being found. There is also the added advantage of a clear division between classes as the data collected comes from sites which specialise in creating headlines of one type.

There approximately 26000 records contained in the dataset and each record in the dataset consists of three attributes:

- **is_sarcastic**: Has a value of 1 if the record is sarcastic and 0 if is not.
- **headline**: The article headline.
- **article_link**: Original news article link.

The dataset is pre-processed before it can be used as input for our logistic regression model. The data is first read in and the **article_link** column is dropped as it is unnecessary due to the presence of the labels. The headlines are then converted to lowercase to make classification easier as capital letters may be recognised as different letters to their lowercase versions. The exclamation mark is then replaced with its word representation, as it may prove useful as a sarcasm indicator as seen in papers in the related section, and all other punctuation is removed from the headlines. Tokenization is then performed on the headlines to separate them into stand-alone words, after this a word stemming process is applied to the headlines to reduce words to their root forms which helps reduce any redundancy which may occur to tense variation. The pre-processed data is then saved and stored in csv format for later use.

Before the main model is run the dataset for the main logistic regression model the train_test_split function from the sklearn library is used to split the dataset into the training set and testing set, the test set consists of 20% of the data while the training set consists of 80% of the data. The training dataset is further split into the the validation set which contains 20% of the data from the initial training data and the final training set which consists of 80% of the data from the initial training data. The datasets are then vectorized using the *TfidfVectorizer* which transforms the headlines to feature vectors for use in our model. The normalization parameters such as the mean and standard deviation for the final training set is calculated, the training set, testing set and validation set are all normalized using the calculated values. As the final step before the data is ready for fitting we append a bias term to all the rows of the training, testing and validation sets.

IV. Methods

A. Description of equations in model

The equations below play a crucial role in our logistic regression and the model as a whole.

$$h_\Theta(x) = \frac{1}{1 + \exp -\Theta^T x} \tag{1}$$

Equation 1 represents the probability that the prediction class/response variable $y = 1$ and makes use of the fact that the sigmoid function exists between 1 and 0. Therefore it is a perfect fit for binary logistic regression.

$$J(\Theta) = \frac{1}{2N} \sum_{n=1}^{N} (h_\theta(x^{(n)}) - y^{(n)})^2 \tag{2}$$

Equation 2 represents the cost function used in our algorithm, it is calculated as the average of the sum of the squared differences between predicted output and actual output values.

$$\theta_k^{(t)} = \theta_k^{(t-1)} - \alpha \frac{1}{N} \sum_{n=1}^{N} (h_\theta(x^{(n)}) - y^{(n)}) x_k^{(n)} \tag{3}$$

Equation 3 represents the update function of the algorithm, which a chosen alpha the function finds the point that minimises the cost function to give us parameters that enable accurate prediction.

B. Description of learning method

The model learns by taking the in the training dataset, a chosen alpha (learning rate), initial parameters and number of iterations as input. While the algorithm has not run the full number of iterations equation 1 is used to calculate the error which is then passed to equation 2 to calculate the cost which is stored. The error value is then utilized in equation 3 to update the theta values of the model. Once the number of iterations goes over the specified value the learning method returns the updated theta values and stored cost values. The theta values are then used for prediction on the training set and validation set to produce the empirical error percentages. A graph is plotted using the stored values from the cost function to show the change in cost as the model ran through its iterations. The process above is repeated for multiple alpha values. The alpha value that produces the lowest error on the validation dataset is then used to fit a model and make predictions on the test dataset. Based on the actual response variable values and the predicted response variable values of the tuned model, the confusion matrix, accuracy score, recall, precision and f1 score are calculated.

A cross validation model setup which uses the learning method is utilized with the tuned hyper parameters to check the overall quality of the model above. A SVM model with built-in methods is also fitted with the dataset for us to compare the performance of our models.

V. Experimental Results and Discussion

The tables and graphs below relate to the main logistic regression model, additional results for other models will only be mentioned for comparison.

TABLE I
Error percentages for different alphas

	ALPHA VALUES			
	0.01	**0.05**	**0.1**	**0.2**
Error train	**10.55**	**8.37**	**7.45**	**6.72**
Error validate	**16.96**	**17.62**	**18.20**	**18.95**

TABLE II
Confusion matrix for alpha=0.01

2495	**511**
441	**1895**

TABLE III
Model statistics for alpha=0.01

Accuracy	**82.18**
Recall	**81.12**
Precision	**78.76**
F1 score	**79.92**

Table 1 represents the error percentages obtained after fitting and prediction of the logistic regression model with varying alphas as seen in the table. A fixed number of iterations of 300 was used during the model fitting to prevent long training times. From the table we can see that as larger alphas are used to fit the model, the error percentage on the training data decreases which means a larger learning rate benefits the model on the training data. The opposite is true for the error percentage on the validation data, it increases as alpha increases, therefore the percentage of error the smallest alpha gives the lowest error for prediction. Based on these results we set alpha as 0.01 for the model and made predictions on our test data.

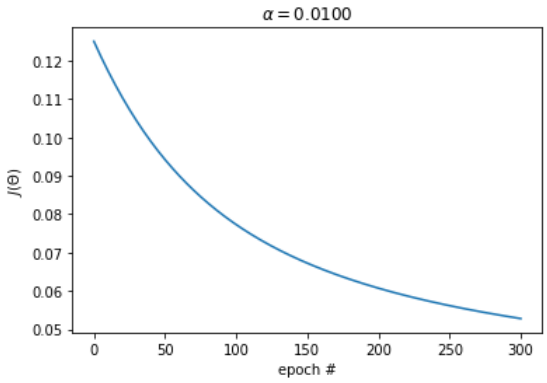

Fig. 1. Cost graph for alpha=0.01

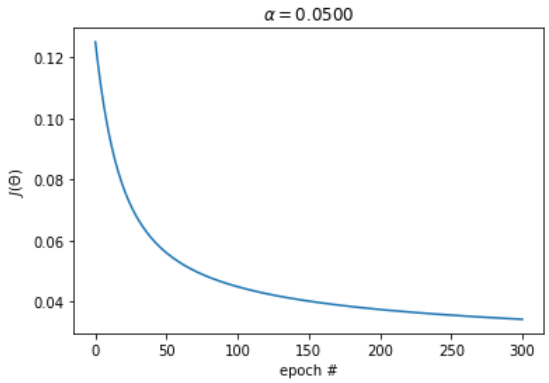

Fig. 2. Cost graph for alpha=0.05

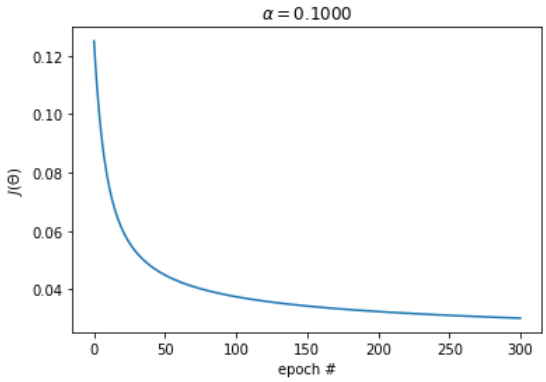

Fig. 3. Cost graph for alpha=0.1

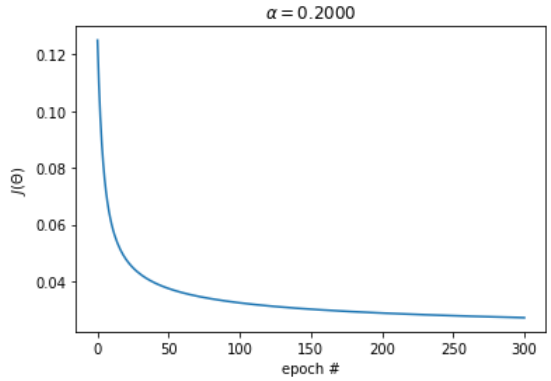

Fig. 4. Cost graph for alpha=0.2

Fig. 1,2,3 and 4 represent the cost graphs obtained while each alpha in table 1 was being used in the logistic regression model. From these four graphs we are able to see that a larger learning rate meant that the model converged faster although due to error rates we can see that a lower alpha produces a lower error percentage during prediction on the validation data.

Table 2 and 3 represent the results of the model prediction on the test data with alpha set as 0.01 and the number of iterations set at 300. The formulas for the statistics can be seen below:

$recall = \frac{TruePositive}{TruePositive+FalseNegative}$,

$precision = \frac{TruePositive}{TruePositive+FalsePositive}$ and

$F1 = 2 * \frac{precision*recall}{precision+recall}$

The accuracy from table 3 is 82.18%, we can see where this percentage came from by looking at the sum of the True positives and True Negative from table 2 over the total of the table. This score is impressive on the test data as it means that a large number of records were classified correctly, as being sarcastic or not. Researchers usually strive for a model with predictive accuracy higher than 80%, therefore the model has good predictive power. This is very similar to the cross validated model of the logistic regression which obtained an error percentage of 16.98% on the test data which means the accuracy of the model is also above 80% and therefore leads us to believe the original model is of good quality. The cross validation model was applied with 5 folds, therefore its error percentage on the test data is an average representation. The SVM model that was fitted has an accuracy of 82.87% which shows that our logistic regression achieves similar results to that of a built-in python model.

Recall is the ratio of true positives over the number of positive examples, our model achieved a recall of 81.12% which indicates that the class (sarcastic) is recognized correctly. Precision represents the ratio of true positives to the total number of predicted examples that are positive, our model has a precision of 78.76 which is high as well, this indicates a reduced number of records are classified as false positives and means the model correctly identifies a large amount of records that are actually sarcastic. The F1 score is approximately 80% which is an additional indicator that the model has decent predictive abilities. Our model achieved results on par with research papers mentioned in our related section and even performed better than some of them, since most of the other models focus on Twitter data which has a lot of noise.

VI. Conclusion

Our model is of good quality, although like [2] it may not perform well in real time analysis. More research needs to carried out on improving this issue as we do not how the model will behave with headlines from another news website, since the headlines in our dataset are from only two websites.

References

[1] M. Bouazizi and T. O. Ohtsuki, "A pattern-based approach for sarcasm detection on twitter," *IEEE Access*, vol. 4, pp. 5477–5488, 2016.

[2] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on twitter: A behavioral modeling approach," in *Proceedings of the eighth ACM international conference on web search and data mining*, 2015, pp. 97–106.

[3] S. G. Shamay-Tsoory, R. Tomer, and J. Aharon-Peretz, "The neuroanatomical basis of understanding sarcasm and its relationship to social cognition." *Neuropsychology*, vol. 19, no. 3, p. 288, 2005.
[4] F. Stringfellow Jr, *Meaning of Irony, The: A Psychoanalytic Investigation*. SUNY Press, 1994.
[5] C. Burfoot and T. Baldwin, "Automatic satire detection: Are you having a laugh?" in *Proceedings of the ACL-IJCNLP 2009 conference short papers*, 2009, pp. 161–164.
[6] J. Tepperman, D. Traum, and S. Narayanan, "" yeah right": Sarcasm recognition for spoken dialogue systems," in *Ninth international conference on spoken language processing*, 2006.
[7] F. Barbieri, H. Saggion, and F. Ronzano, "Modelling sarcasm in twitter, a novel approach," in *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2014, pp. 50–58.
[8] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 704–714.
[9] R. Misra and P. Arora, "Sarcasm detection using hybrid neural network," *arXiv preprint arXiv:1908.07414*, 2019.