Sarcasm Detection on Twitter Data Using R and Python

V. Haripriya, Poornima G. Patil, and T. V. Anil Kumar

Abstract A challenging task in Opinion Mining is to detect sarcasm. Sarcasm is a way of telling things in a positive way where the intended meaning is negative. Sarcastic texts lead to wrong classification of data due to the usage of positive words to convey negative emotions. It is important to detect sarcasm on social media data because there is no face-to-face conversation happening to understand the context. When this data is taken for sentiment analysis and decision-making process it leads to wrong inferences. The proposed work classifies the dataset into sarcastic and non-sarcastic using machine learning algorithms and the same has been implemented using R and Python for finding the difference in accuracy. The dataset has been collected from Twitter. Naïve Bayes algorithm is used for the classification. The result obtained from sarcasm detection on social media data will be helpful in areas such as sentiment analysis, natural language processing, opinion mining, business-related decision making, and so on. The result of the proposed method classified the given data into sarcastic and non-sarcastic with an accuracy of 0.91 and 0.88 using R and Python respectively.

Keywords Opinion mining · Sarcasm detection · Sentiment analysis · Naïve Bayes

1 Introduction

The word Sarcasm is derived from the Latin word which means "to tear flesh", and it has been called "hostility disguised as humor" and generally used to make fun of others by some people (Teh et al. 2018). Presence of sarcasm can completely change the meaning of a sentence. Hence it is important to identify the sarcasm in opinion mining and sentiment analysis. Sarcasm is considered to be an important aspect in the reviews because of the absence of face-to-face conversation. Since the growth of internet usage is getting higher over time and the volume of data

V. Haripriya (✉) · P. G. Patil
Visvesvaraya Technological University, Belgaum 590018, India

T. V. Anil Kumar
LEAD College of Management, Palakkad 678009, India

M. Mukherjee et al. (eds.), *Advances in Medical Physics and Healthcare Engineering*, Lecture Notes in Bioengineering, https://doi.org/10.1007/978-981-33-6915-3_45

generation is huge, the process of data analysis becomes a tedious task. Sarcasm leads to an additional complication in the text analysis process. Social media sites act as a platform for the users to express their views and opinions on various topics and events. When a customer tweets “Great multipurpose phone! Can be used as a phone and as a hot iron as well”, a review taken from a branded mobile contains positive sentiment words but the review is actually negative as it being mocked for over-heating. This results in the review to be wrongly classified into positive sentiment which will directly or indirectly affect the decision making for business development. This is why detection of sarcasm is important. Selecting and describing a set of features for identifying sarcasm at a linguistic level, especially in short texts created in social media such as Twitter postings or “tweets” is being followed widely. In the context of text classification, sarcasm detection is an important factor since it has many implications on various fields such as health, sales, politics, and many more (Sarsam et al. 2020). Sarcasm has a major effect on Sentiment analysis. But most of the research works ignored sarcasm detection due to its difficulty in finding. Numerous researches are being done in order to detect sarcasm with the help of the usage of words and various features in text data. There are also methods where models are built in order to extract sentiments and contextual information from the data. Supervised and Unsupervised approaches can be used for detecting sarcasm. Researches have been done using a lexicon-based, rule-based approach, corpus-based approach, statistical-based approach, and various machine learning approaches. The proposed work uses Naïve Bayes classification, a machine learning approach to detect sarcasm in tweets.

2 Literature Survey

Numerous studies and researches had been conducted on Sentiment Analysis. It was during the early 1990s that the research was started in the field of sentiment analysis. The term sentiment analysis along with opinion mining was first introduced in the year 2003 by Dave et al. (2003) during this time, the work was very much limited only to subjective detection, sentiment adjectives, and interpretation of metaphors and within 15 years the research interests in this area have increased manifold (Pang and Lee 2008; Kumar and Teeja 2012). Some remarkable studies mentioned that sarcasm in a sentence by conflict and contrast of sentiment polarity (Camp 2012; Riloff et al. 2013; Joshi et al. 2017). Kumar and Garg (2019) suggested that detection of sarcasm in natural language text is a well-accepted problem in the area of sentiment analysis.

Parveen and Pandey (2016) discussed the extraction process of sentiment from Twitter and they have analyzed the tweets to provide predictions on business intelligence using Naïve Bayes algorithm. Authors have used Hadoop Framework and worked on movie data from Twitter. The reviews of movies have been classified as positive, negative, and neutral sentiments. The proposed work proves that performance of the Naive Bayes algorithm increases by converting the emoticons into its equivalent word. A novel approach has been used (Khan et al. 2014) to classify the tweets into positive, negative, and neutral feelings using emoticon classifier, Bag-of-words, and SentiWordnet classifier and obtained increased classification accuracy. Authors (Samonte et al. 2018; Farías et al. 2018) mentioned Naive Bayes, Maximum Entropy, and Support Vector Machine are commonly used as the machine learning algorithms for the detection of sarcasm. Affective and structural features are employed (Mukherjee and Bala 2017) to predict irony with conventional machine learning classifiers such as Decision Tree, SVM, and Naïve Bayes. In a follow-up study by Farías et al. (2016) have used a knowledge-based k-NN classifier with a feature set that could capture a broad range of linguistic phenomena namely structural and emotional. Apart from machine learning algorithm, many research works are done using Deep learning networks. Saha et al. (2017) used Naïve Bayes and Support Vector Machine (SVM) classifiers to classify the data for sarcasm detection and also aim to differentiate between the accuracy, precision, recall, and F-score of Naïve Bayes and SVM classifier. Zhang et al. (2011) classified sentiment using Naïve Bayes and SVM for restaurant reviews written in Cantonese. The highest accuracy reported was 95.67% using Naïve Bayes. Kiilu et al. (2018) developed an approach for detecting and classifying hateful speech that uses content produced by self-identifying hateful communities from Twitter. Results from their experiments showed that Naive Bayes classifier achieved significantly better performance than existing methods in hate speech detection algorithms. The work is done by Van Hee et al. (2018) used a combination of lexical, semantic, and syntactic features and were implemented using Support Vector Machine Classifier and results showed that it outperformed Long short-term memory, deep neural network approaches. Potamias et al. (2020) employed advanced deep learning methodologies to tackle the problem of identifying the Figurative language forms. The performance was measured using the devised hybrid neural architecture and experimented with various datasets and contrasted with other appropriate state-of-the-art methodologies and systems. Irsoy and Cardie (2014) applied the deep Recurrent Neural Network (RNN) to the task of opinion expression extraction formulated as a token-level sequence labeling task. Experimental results show that deep, narrow RNNs outperform traditional shallow, wide RNNs with the same number of parameters.

Sarcasm detection becomes difficult without having adequate knowledge of the "context" of the situation, the particular topic, and the environment (Kumar and Garg 2019). Numerous researches are being done in order to detect sarcasm with the help of usage of the words and various features in text data. There are also methods where models are built in order to extract sentiments and contextual information from the data. Re-searches have been taken place using lexicon-based, rule-based approach, corpus-based approach, statistical-based approach, and various machine learning approaches. The recent work concentrates more on Deep learning approaches (Eke et al. 2020).

Though many researches have been done on Sarcasm detection, a comparison of accuracy level and the time required to complete the process using various tools like R and Python have not been done so far. Thereby the proposed work implemented

Naive Bayes algorithm for the dataset using R and Python. Moreover, this work helps researchers to get an idea of the steps involved in the sarcasm detection process.

3 Methodology

3.1 Data Collection and Preprocessing

The dataset has been collected from Twitter using Twitter API. The collected data is noisy and it cannot be directly fed into the classifier as it will affect the working of the classification algorithm. RE (Regular Expressions) and NLTK (Natural Language Tool Kit) packages in Python were used to clean the data by removing new lines and tabs, punctuations, hashtagged words, emoticons, and emojis and the stop words. In total 1992 preprocessed tweets have been parsed into the system.

3.2 Training and Testing Phase

The collected tweets were divided into training and testing set. In the proposed work the dataset has been split into 70:30 ratio. The purpose of the training dataset is to provide the algorithm with "ground truth" data whereas the test dataset is used to check how well the algorithm was trained with the provided training dataset.

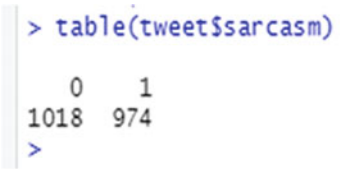
3.3 Implementation

The proposed work used Naïve Bayes approach since it is one among the well-known classification algorithms and it works based on Bayes' Theorem. This algorithm makes use of conditional probability to predict the likelihood of future occurrence of events based on their historical information.

Bayes' Theorem is mentioned as:

$$P(A|B) = P(A)P(B|A)P(B) \tag{1}$$

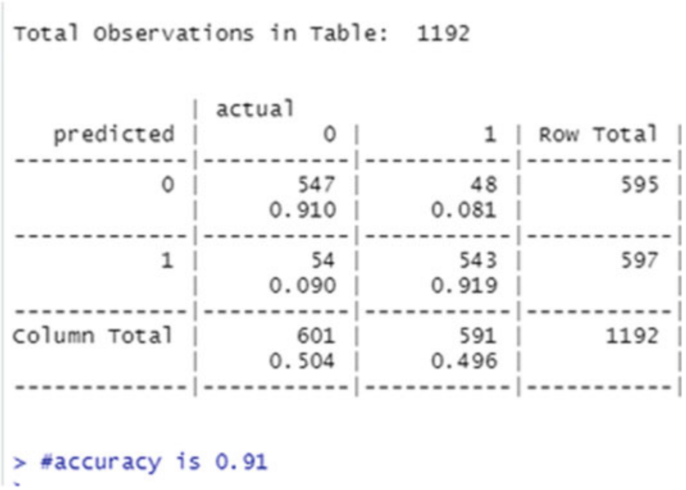
The implementation has been done using R programming and Python to check the difference in accuracy. Various libraries namely Readxl, tm, wordcloud, E1071, gmodels, and naiveBayes have included in R studio whereas Nltk and stopwords were used for implementation using python.



```
> table(tweet$sarcasm)

   0    1
1018  974
>
```

Fig. 1 Classification of sarcastic and non-sarcastic tweets



```
Total Observations in Table:  1192


             | actual
   predicted |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       547 |        48 |       595 |
             |     0.910 |     0.081 |           |
-------------|-----------|-----------|-----------|
           1 |        54 |       543 |       597 |
             |     0.090 |     0.919 |           |
-------------|-----------|-----------|-----------|
Column Total |       601 |       591 |      1192 |
             |     0.504 |     0.496 |           |
-------------|-----------|-----------|-----------|


> #accuracy is 0.91
```

Fig. 2 Evaluation of model performance

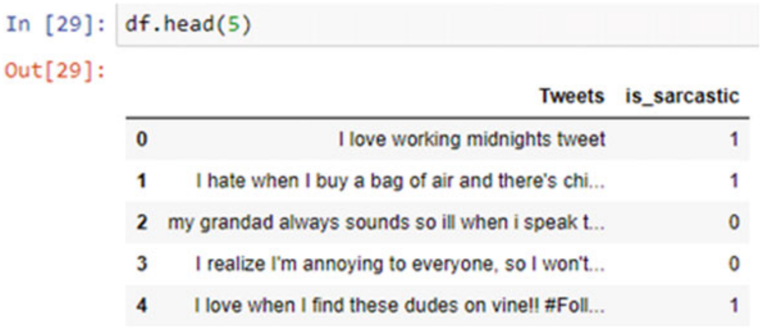
4 Results and Discussion

4.1 Implementation Using R

Naïve Bayes approach is used as the classification algorithm and with the support of various libraries the model has been implemented using R Studio. The given data has been classified as sarcastic and non-sarcastic tweets where the value 0 indicates non-sarcastic and the value 1 represents sarcastic and the same has been depicted in Fig. 1 and the performance of the model has been evaluated and the results obtained are represented in Fig. 2 respectively.

4.2 Implementation Using Python

Dataset has been passed to the Naïve Bayes approach to check whether the given data is sarcastic or non-sarcastic. Nltk and stopwords were used for implementation using python. MultinominalNB has been used for the classification with discrete features

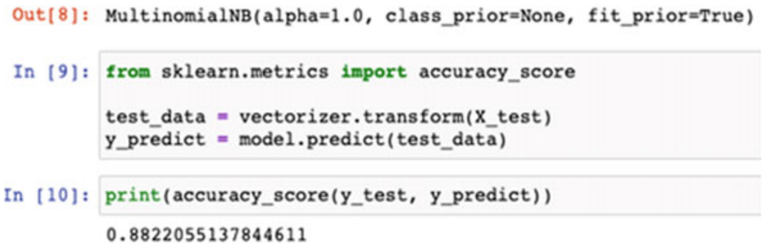


```
In [29]: df.head(5)

Out[29]:
```

	Tweets	is_sarcastic
0	I love working midnights tweet	1
1	I hate when I buy a bag of air and there's chi...	1
2	my grandad always sounds so ill when i speak t...	0
3	I realize I'm annoying to everyone, so I won't...	0
4	I love when I find these dudes on vine!! #Foll...	1

Fig. 3 Classification of sarcastic and non-sarcastic tweets



```
Out[8]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

In [9]: from sklearn.metrics import accuracy_score

        test_data = vectorizer.transform(X_test)
        y_predict = model.predict(test_data)

In [10]: print(accuracy_score(y_test, y_predict))

         0.8822055137844611
```

Fig. 4 Evaluation of model performance

such as word counts for classifying the text. Training and testing data has been split in the ratio of 70:30 where 70% of data has been passed for training the model whereas 30% of data has been passed for testing the model. Tweets have been classified as sarcastic and non-sarcastic and are represented in Fig. 3 and the accuracy obtained was 0.88 and it is depicted in Fig. 4.

5 Conclusion

Sarcasm detection is considered as the challenging factor in sentiment analysis since it gives a gap between the intended meaning and the literal meaning of the sentences. This ambiguity poses challenges to the industries in making proper decisions. By doing sentiment analysis on the social media data, that gives a tremendous change in their way of making decisions. R and Python are open-source and widely used in the analytical process. Statistical analysis is the major purpose of using R whereas data science is the core idea of Python. Python is considered as the most robust language than R. However, R has more built-in analysis for summary statistics and Python works on the basis of packages. The main implications of the proposed work are to help to take decisions based on the reviews or comments written on social media platforms. As future research semantic tools can be used to detect sarcasm and an ontology model can be created to classify the texts accurately by reducing the problem of wrong classification of texts.

References

Camp E (2012) Sarcasm, pretense, and the semantics/pragmatics distinction. Noûs 46(4):587–634

Dave K, Lawrence S, Pennock DM (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th international conference on World Wide Web, pp 519–528

Eke CI, Norman AA, Shuib L, Nweke HF (2020) Sarcasm identification in textual data: systematic review, research challenges and open directions. Artif Intell Rev 53(6):4215–4258

Farías DIH, Patti V, Rosso P (2016) Irony detection in Twitter: the role of affective content. ACM Trans Internet Technol (TOIT) 16(3):1–24

Farías DIH, Montes-y-Gómez M, Escalante HJ, Rosso P, Patti VA (2018) Knowledge-based weighted KNN for detecting irony in Twitter. In: Mexican international conference on artificial intelligence. Springer, Cham, pp 194–206

Irsoy O, Cardie C (2014) Opinion mining with deep recurrent neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 720–728

Joshi A, Bhattacharyya P, Carman MJ (2017) Automatic sarcasm detection: a survey. ACM Comput Surv (CSUR) 50(5):1–22

Khan FH, Qamar U, Javed MY (2014) Sentiview: a visual sentiment analysis framework. In: International conference on information society (i-Society 2014), pp 291–296. IEEE

Kiilu KK, Okeyo G, Rimiru R, Ogada K (2018) Using Naïve Bayes algorithm in detection of hate tweets. Int J Sci Res Publ 8(3)

Kumar A, Garg G (2019) Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. J Ambient Intell Hum Comput 1–16

Kumar A, Teeja MS (2012) Sentiment analysis: a perspective on its past, present and future. Int J Intell Syst Appl 4(10):1

Mukherjee S, Bala PK (2017) Detecting sarcasm in customer tweets: an NLP based approach. Ind Manag Data Syst 117(6):1109–1126

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135

Parveen H, Pandey S (2016) Sentiment analysis on Twitter Data-set using Naive Bayes algorithm. In: 2016 2nd international conference on applied and theoretical computing and communication technology (ICATCCT). IEEE, pp 416–419

Potamias RA, Siolas G, Stafylopatis AG (2020) A transformer-based approach to irony and sarcasm detection. Neural Comput Appl 1–12

Riloff E, Qadir A, Surve P, De Silva L, Gilbert N, Huang R (2013) Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 704–714

Saha S, Yadav J, Ranjan P (2017) Proposed approach for sarcasm detection in Twitter. Indian J Sci Technol 10(25):1–8

Samonte MJC, Dollete CJT, Capanas PMM, Flores MLC, Soriano CB (2018) Sentence-level sarcasm detection in English and Filipino tweets. In: Proceedings of the 4th international conference on industrial and business engineering, pp 181–186

Sarsam SM, Al-Samarraie H, Alzahrani AI, Wright B (2020) Sarcasm detection using machine learning algorithms in Twitter: a systematic review. Int J Market Res 1470785320921779

Teh PL, Ooi PB, Chan NN, Chuah YK (2018) A comparative study of the effectiveness of sentiment tools and human coding in sarcasm detection. J Syst Inf Technol

Van Hee C, Lefever E, Hoste V (2018) Exploring the fine-grained analysis and automatic detection of irony on Twitter. Lang Resour Eval 52(3):707–731

Zhang Z, Ye Q, Zhang Z, Li Y (2011) Sentiment classification of Internet restaurant reviews written in Cantonese. Expert Syst Appl 38(6):7674–7682