Sarcasm Detection of Amazon Alexa Sample Set

Avinash Chandra Pandey, Saksham Raj Seth and Mahima Varshney

Abstract Sentiment analysis using collection of positive, negative score of a word has been one of the most researched topics in Data Mining. This kind of analysis is more prominent based on the content available on social media like comments on Facebook, tweets on Twitter, and the count goes on. Sarcasm can be understood as irony but it is a text spoken in such a manner that evokes laughter and humor. It is a type of sentiment where people express their negative feelings using positive or intensified positive words in the text. While speaking, people often use heavy tonal stress and certain gestures clues like rolling of the eyes, hand movement, etc., to reveal sarcasm. In this paper, NLTK has been used which is a Python toolkit to harness the power of generating information from the huge text datasets available. Sampled data from Amazon Alexa has been collected which is further processed using SentiWordNet 3.0 and TextBlob to remove noise and irrelevant data. Thereafter, Gaussian naive Bayes algorithm along with TextBlob has been used to detect sarcasm in dataset. The performance of the proposed method is compared with naïve Bayes, decision tree, and support vector machine. From the experimental results, effectiveness of the proposed method is observed.

Keywords SentiWordNet 3.0 · TextBlob · Semi-supervised classification · NLTK POS vector · POS-Tag · Naïve Bayes · Capitalization

A. C. Pandey · S. R. Seth (✉) · M. Varshney
Jaypee Institute of Information Technology, Noida, India
e-mail: saksham2801@gmail.com

A. C. Pandey
e-mail: avinash.pandey@jiit.ac.in

M. Varshney
e-mail: mahimavarshney011@gmail.com

B. S. Rawat et al. (eds.), *Advances in Signal Processing and Communication* ,
Lecture Notes in Electrical Engineering 526,
https://doi.org/10.1007/978-981-13-2553-3_54

1 Introduction

In the present-day world where humans are having conflicting emotions, it is a tedious task to analyze their sentiments. Sarcasm requires shared knowledge between speaker and the listener [1]. Detection of sarcasm in text is difficult because gestural and tonal clues are missing. Many machine learners collect their dataset from social texts to detect sarcasm, especially in tweets [1]. We used the dataset provided by the Amazon Alexa's sample set to apply machine learning algorithms. A machine learning algorithm is attempted to design to detect sarcasm in text. Naive Bayes, one-class SVM and Gaussian kernel are few algorithms commonly used to perform the same task [2].

Semi-supervised sarcasm is identified on two different datasets: a collection of millions of tweets collected from Twitter, and a collection of millions of product reviews from Amazon [3]. On Twitter a common form of sarcasm exists in a form where a positive sentiment contradicted with a negative situation. For example, many sarcastic tweets include a positive sentiment, such as "love" or "enjoy", followed by an expression that describes an undesirable activity or state (e.g., "taking exams" or "being ignored") [4].

Sarcasm changes the polarity of an apparently positive or negative statement into its contradictory statement. A corpus of sarcastic messages on Twitter is created by many authors on whom determination of the sarcasm of each message has been made by its author. These corpuses are used as a reliable benchmark to compare sarcastic expressions in Twitter. Many authors also investigated the impact of lexical and pragmatic factors for discovering sarcastic statements. Sarcastic statements are difficult to identify. Therefore, we compare the performance of machine learning techniques and human judges on this task to find who is performing better. Perhaps unsurprisingly, neither the human judges nor the machine learning techniques [5–7] perform very well [8]. There are many computational approaches for sarcasm detection using lexical cues has been given [9].

Many properties [10–14] were explored while finding sarcasm in text like theories of sarcasm, syntactical properties [15], lexical feature [16, 17], etc. [4, 18, 19]. Model's accuracy can be improved after finding positive and negative works, which can be done using bag-of-words. Accuracy increases for feature extraction by the use of bag-of-words [12]. The experimental results depict that the proposed method outperforms the existing methods. The rest of the paper is organized as follows: Sect. 2 describes the proposed method. Section 3 discusses experimental results and Sect. 4 concludes the paper.

2 Proposed Work

In this research paper, we performed various operations to build our model. SentiWordNet 3.0 dictionary is preprocessed and transformed to form a map, which

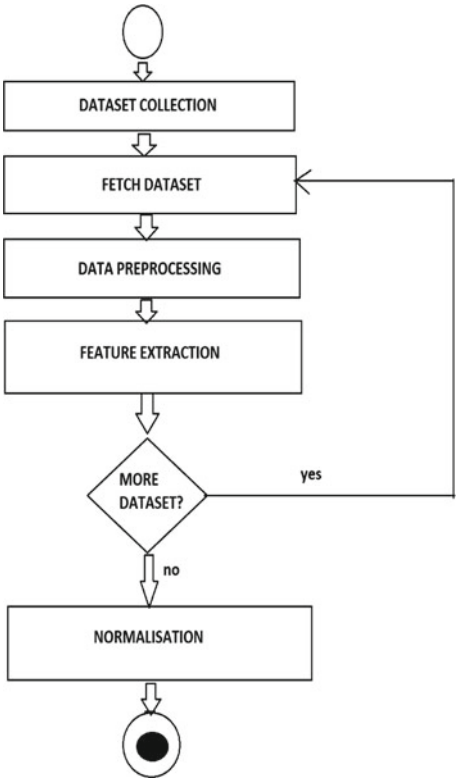

Fig. 1 Flowchart shows the process of data processing, features extraction to detect sarcasm

contains a key and value from the dictionary, where key contains the POS tags and synsets of the SentiWordNet dictionary and value contains the mean of positive and negative values of the respective words in the SentiWordNet dictionary. We used this map to calculate to sentiment of the provided textual data. The complete steps of proposed method have been shown in Fig. 1.

The aim of TextBlob is to provide access to common text processing operations. Polarity and Subjectivity are the main factors of Python library, i.e., TextBlob. TextBlob objects can be treated as Python library to do Natural Language Processing. On the provided textual data, polarity and subjectivity are calculated by the TextBlob objects, to improve the sentiment score. Above two methods were very useful and improvement in accuracy was up to 5–7%. Apart from these two methods, we implemented Vectorization method.

In which a vector was created to store the count of Nouns, Adverbs, Adjectives, and Verbs in the provided textual data. This method was implemented with the help of POS-TAG (Part-Of-Speech Tagging), a very impressive method in the Python library in NLTK (Natural Language Toolkit). NLTK library deals with the textual data and simplifies work for Python programmers. Method POS-TAG returns a list

Table 1 Accuracy of the existing method and the proposed method

Sr. No.	Methods	Accuracy (%)
1	Naive Bayes	65.35
2	Decision tree	65.78
3	SVM	69.37
4	Proposed method	70.96

of each word from the provided textual data, with the tags of Nouns, Verbs, Adverbs, Adjectives, etc.

To improve our accuracy for about 2–3%, we implemented a technique called Capitalization. In which the focus is given on the words which are Capital, so that we can detect the words which are to be focused to be spoken. When we provide a textual data, we have no idea which word is given stress on. This was a very impressive technique to judge the textual data's sense.

A matrix was created for the whole dataset containing the features extracted from the above techniques and final step taken was to apply naive Bayes Algorithm. The naive Bayes is used as a baseline for text categorization. The classifier makes the naïve assumption that the independence occurs between all the features. The classifier is applied from Bayes theorem. Its simplicity makes it a popular machine learning classifier.

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}$$

On a whole, after the application of all these great techniques, an accuracy of 70.96% was obtained.

3 Experimental Results

The performance of the proposed method has been tested on sarcasm dataset and its accuracy is also compared with naïve Bayes, decision tree, and SVM. From Table 1, it is easily observed that the proposed method outperforms the exiting method. Moreover, histogram for accuracy is also plotted in Fig. 2. From Fig. 2, the effectiveness of the proposed method can be easily observed.

The above histogram shows the accuracy rate variation for executing the same model for three times.

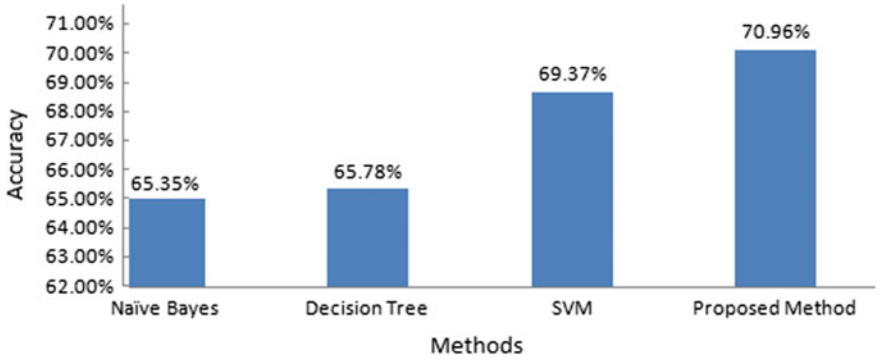

Fig. 2 Comparison of various models on sarcasm dataset

4 Conclusion

Automatic sarcasm detection is a formidable task. This paper offers novel naïve Bayes method to detect sarcasm in Amazon Alexa dataset [20]. The dataset is divided into training and test dataset using cross-validation techniques. The quality of features/attributes extracted from the training dataset affects the performance of the technique. Therefore, SentiWordNet and TextBlob have been used to extract important features from dataset and the model is trained using those features. The test dataset is tested using Gauss-based naïve Bayes method and three baseline methods namely; naïve Bayes, decision tree, and support vector machine. From the experimental results, it is found that the proposed method outperforms the baseline methods.

Sarcasm is closely related to language- or culture-specific traits. Future approaches to identity sarcasm in new languages can benefit to identify such traits.

References

1. Bharti, S.K., Vachha, B., Pradhan, R.K., Babu, K.S., Jena, S.K.: Sarcastic sentiment detection in tweets streamed in real time: a big data approach. Digital Communications and Networks **2**(3), 108–121 (2016)
2. Peng, C.-C., Lakis, M., Pan, J.W.: Detecting Sarcasm in Text: An Obvious Solution to a Trivial Problem (2015)
3. Dmitry, D., Tsur, O., Rappoport, A.: Semi-supervised recognition of sarcastic sentences in twitter and amazon. In: Proceedings of the fourteenth conference on computational natural language learning, pp. 107–116. Association for Computational Linguistics (2010)
4. Ellen, R., Qadir, A., Surve, P., De Silva, L., Gilbert, N., Huang, R.: 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 (2013)
5. Pandey, A.C., Pal, R., Kulhari, A.: Unsupervised data classification using improved biogeography based optimization. Int. J. Syst. Assur. Eng. Manag. 1–9

6. Pandey, A.C., Rajpoot, D.S., Saraswat, M.: Data clustering using hybrid improved cuckoo search method. In: 2016 Ninth International Conference on Contemporary Computing (IC3), pp. 1–6. IEEE (2016)
7. Pal, R., Avinash Pandey, H.M., Saraswat, M.: BEECP: Biogeography optimization-based energy efficient clustering protocol for HWSNS. In: 2016 Ninth International Conference on Contemporary Computing (IC3), pp. 1–6. IEEE (2016)
8. González-Ibánez, R., Muresan, S., Wacholder, N.: 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, vol. 2, pp. 581–586. Association for Computational Linguistics (2011)
9. Forslid, E., Niklas, W.: Automatic Irony-and Sarcasm Detection in Social Media (2015)
10. Bamman, D., Smith. N.A.: Contextualized sarcasm detection on Twitter. In: ICWSM, pp. 574–577 (2015)
11. Pandey, A.C., Rajpoot, D.S., Saraswat, M.: Twitter sentiment analysis using hybrid cuckoo search method. Inf. Process. Manag. **53**(4) 764–779 (2017)
12. Wicana, S.G., İbisoglu, T.Y., Yavanoglu, U.: A Review on sarcasm detection from machine-learning perspective. In: 2017 IEEE 11th International Conference on Semantic Computing (ICSC), pp. 469–476. IEEE (2017)
13. Dave, A.D., Desai, N.P.: A comprehensive study of classification techniques for sarcasm detection on textual data. In: International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 1985–1991. IEEE (2016)
14. Rajadesingan, A., Zafarani, R., Liu, H.: Sarcasm detection on twitter: a behavioral modeling approach. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 97–106. ACM, (2015)
15. Mishra, A., Kanojia, D., Seema N., Kuntal D., Bhattacharyya, P.: Harnessing Cognitive Features for Sarcasm Detection (2017). arXiv:1701.05574
16. Sharada, A., Krishna, P.P.: Sentiment Mining: an approach for Hindi reviews. Algorithms (2017)
17. Forslid, E., Wikén, N.: Automatic Irony-and Sarcasm Detection in Social Media (2015)
18. Detection Ratcliffe, C., Griffith, J., A Machine Learning Approach to Automatic Sarcasm. National University of Ireland, Galway
19. Joshi, A., Kanojia, D., Bhattacharyya, P., Carman, M.J.: Sarcasm Suite: a browser-based engine for sarcasm detection and generation. In: AAAI, pp. 5095–5096 (2017)
20. Amazon Alexa dataset, http://curtis.ml.cmu.edu/w/courses/index.php/Amazon_Dataset_for_Sarcasm