Journal of Engineering Sciences
Vol 12, Issue 04, April/2021
ISSN NO:0377-9254

An Extension of Lexicon Algorithm for Sarcasm Detection from Online Social Networks

Dr. Abdul Ahad[1], Masadi Praveen Kumar[2], Podduturi Vinutha[2], V. Bhanuchander Reddy[2]

[1]Professor, [2]UG Scholar, [1,2]Department of CSE

[1,2]Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Hyderabad, Telangana

Abstract- Lexicon algorithm is used to determine the sentiment expressed by a textual content. This sentiment might be negative, neutral or positive. It is possible to be sarcastic using only positive or neutral sentiment textual contents. Hence, lexicon algorithm can be useful but yet insufficient for sarcasm detection. It is necessary to extend the lexicon algorithm in order to come out with systems that would be proven efficient for sarcasm detection on neutral and positive sentiment textual contents. In this paper, two sarcasm analysis systems both obtained from the extension of the lexicon algorithm have been proposed for that sake. The first system consists of the combination of a lexicon algorithm and a pure sarcasm analysis algorithm. The second system consists of the combination of a lexicon algorithm and a sentiment prediction algorithm.

Keywords: Lexicon algorithm, sentiment and sarcasm.

I. INTRODUCTION

Communication is the process of exchanging information. As time goes by, many ways and platforms of communication are being developed. Since the industrial revolution, the original way of communicating; face-to-face communication has been used as a model to develop the various ways of communicating known to date. Transposing the principles and codes of the natural face-to-face communication to today's online communication is a major challenge for developers. Sarcasm is the communication practice that consists of meaning the opposite of what is said in order to mock or insult someone [1]. Sarcasm makes use of positive lingual contents in order to convey a negative message. Different types of approaches have been developed in order to implement sarcasm detection on online communication platforms. However the levels of efficiency of these approaches have been the principal worries of developers. In this paper, propositions are made on how the lexicon algorithm can be extended in order to come out with systems that would be proven more efficient for sarcasm detection on textual contents.

II. RELATED WORK

In [2], Palanisamy, Yadav, & Elchuri used a lexicon based approach to discover sentiments. They used preprocessing steps such as stemming, emoticon detection and normalization, exaggerated word shortening and hash tag detection. After the preprocessing, the lexicon-based system classified the tweets as positive or negative based on the contextual sentiment orientation of the words. In [3], Jurek, Mulvenna, & Bi developed an algorithm that consisted of two key components, namely sentiment normalization and evidence-based combination function, which had been used in order to estimate the intensity of the sentiment rather than positive/negative label and to support the mixed sentiment classification process. In [4], Kiilu, Okeyo, Rimiru, & Ogada 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 with precision, recall, and accuracy values of 58%, 62%, and 67.47% respectively. In [5], Rathan & Suchithra proved that by associating a combined approach of hyperbole, emoticons,

lexical analysis and contrast to sarcasm analysis, better accuracy could be achieved better when compared to usage of only linguistic features.

PROBLEM STATEMENT

Lexicon algorithm makes use of a sentiment lexicon. A sentiment lexicon is a collection of known and precompiled sentiment terms [8]. Lexicon algorithm computes the polarity of each term in a textual content in order to deduce the sentiment expressed through that textual content [9]. This sentiment can either be negative, neutral or positive. However, lexicon algorithm is insufficient for sarcasm detection as it limits itself to give the polarity of a textual content without being able to specify whether that textual content is a sarcastic one or not.

III. PROPOSED WORK

In this paper, the lexicon algorithm has been extended in two ways so as to generate two systems that could be more efficient for sarcasm analysis, especially on neutral and positive sentiment textual contents.

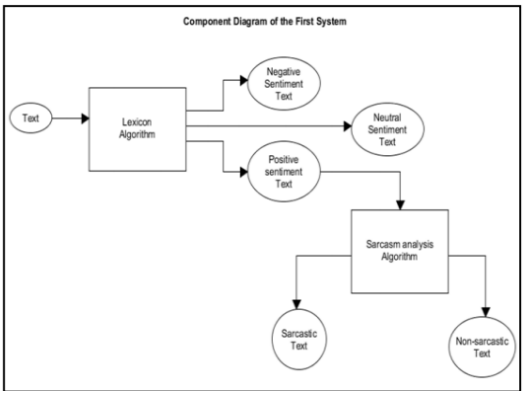

Fig. 1. Component diagram of the first system. An overview of the arrangement of components of the first system.

A. First system

The first system (Fig. 1) is the combination of a lexicon algorithm and a pure sarcasm analysis algorithm. This system takes textual contents as input. These contents could be from various social media platforms like Twitter or Facebook. The textual contents are parsed into the lexicon algorithm for polarity computation. Then the positive sentiment contents are parsed into the pure sarcasm analysis algorithm for sarcasm detection. The final output of this system is a list of sarcastic and non-sarcastic lingual contents.

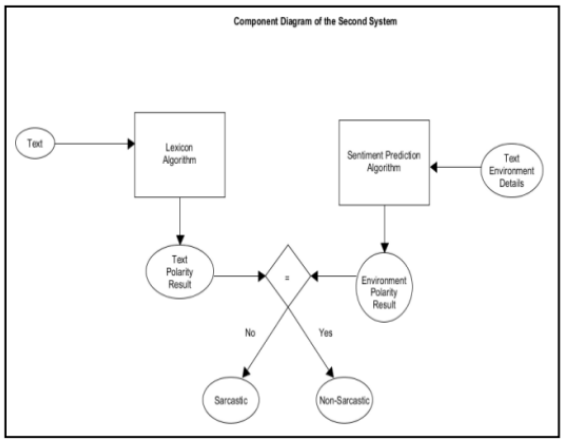

Fig. 2. Component diagram of the second system. An overview of the arrangement of the components of the second system

B. Second system

The second system is the combination of a lexicon algorithm and a sentiment prediction algorithm. The lexicon algorithm is used here the same way as in the first system. The sentiment prediction algorithm consists of a mechanism that can predict the sentiment of a textual content that would be made under a specific environment. The sentiment prediction algorithm takes as input the details of the environment under which a lingual content would be made notably the state of the context, the author's knowledge of the domain he/she would talk about, the author's level of education, the author's personality, the author's relationship with his/her interlocutor. The sentiment prediction algorithm processes these details and predicts the sentiment of the textual content that would be formed under that environment. The results from both the algorithms are compared. In Case the results are different for a textual content, this later is classified as sarcastic else it is classified as non-sarcastic (Fig. 2).

These environment details are processed based on a training data set that was formed as follow:

- Each environment detail had a polarity. The polarity could be negative, neutral or positive (TABLE I).
- An AND operation was performed in between the details polarities of each textual content environment to predict the sentiment of the textual content that would be made under each environment. The training data set of the sentiment prediction algorithm was formed of the environment details polarities and their predicted sentiment (TABLE II).

Table 1. Environment details and their polarity values

Environment Details	Polarity		
	Negative (-)	**Neutral (0)**	**Positive (+)**
State of the context (c)	Tensed (c-)	Neutral (c0)	Calm (c+)
Author's knowledge of the domain (k)	Novice (k-)	Fair (k0)	Good (k+)
Author's level of education (le)	Primary (le-)	Secondary (le0)	University (le+)
Author's personality (ap)	Pessimist (ap-)	Realistic (ap0)	Optimist (ap+)
Author's relationship with his/her interlocutor (ri)	Public (ri-)	Just know (ri0)	Close (ri+)

Table 2: Training data set

Environment Details Polarities	Predicted Sentiment
c+ k- le+ ap+ ri+	N (Negative)
c+ k+ le+ ap+ ri+	P (Positive)
c- k- le- ap- ri-	N (Negative)
c- k- le+ ap+ ri-	N (Negative)
c+ k+ le- ap- ri+	P (Positive)
c+ k- le- ap- ri-	P (Positive)
c+ k- le0 ap+ ri+	Ne (Neutral)
c+ k0 le- ap- ri-	Ne (Neutral)
c0 k- le+ ap0 ri+	Ne (Neutral)

C.STRUCTURE OF THE ALGORITHMS

All the algorithms used in this paper have been derived from the classic Naïve Bayes algorithm. Naïve Bayes algorithm was named after Rev. Thomas Bayes [10]. This algorithm makes use of conditional probability to predict the likeness of future occurrence of events based on their historical information [10]

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

Where A and B are events and $P(B) \neq 0$ [11].

- $P(A|B)$ is a conditional probability: the likelihood of event A occurring given that B is true [11].
- $P(B|A)$ is also a conditional probability: the likelihood of event B occurring given that A is true [11].
- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently for each other; this is known as the marginal probability [11].
- Naïve Bayes is mainly used for classification purposes. It is an algorithm that discriminates different objects based on certain features [11]. This algorithm is built after the Bayes theorem which assumes that all features within a class are independent from one another and that is why it is known as 'naive' [12].

In TABLE II, an AND operation is performed in between the polarities of each environment details to determine the sentiment of the potential textual content that would be formed under each environment. Naïve Bayes is mainly used for classification purposes. It is an algorithm that discriminates different objects based on certain features [11]. This algorithm is built after the Bayes theorem which assumes that all features within a class are independent from one another and that is why it is known as 'naive' [12]. There are several types of Naïve Bayes models [12][13]:

- Gaussian Naïve Bayes: where the predictors take up continuous value and are not discrete.
- Bernouilli Naïve Bayes: where the parameters of the predictors are Boolean values; 'yes', 'no', '1' or '0'.
- Multinomial Naïve Bayes: it is the generalization of Bernouilli where the

features used by the classifier are the frequency of objects being processed. Multinomial Naïve Bayes is the model used in this paper

The steps of the Naïve Bayes algorithm can be resumed to the following [12]:

- Convert the data set into a frequency table.
- Create a likelihood table.
- Use Naive Bayesian equation to calculate the posterior probability for each class.

IV. TESTS, RESULTS AND DISCUSSION

A. Test and Results of the First System

In this experiment, fifty seven preprocessed textual contents, collected from Twitter have been parsed in the system. These contents are then categorized as negative, neutral and positive (Fig. 3) by the lexicon algorithm. The positive sentiment contents are later categorized as sarcastic and non-sarcastic (Fig. 4) by the pure sarcasm analysis algorithm.

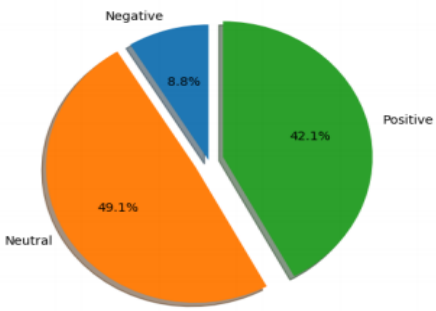

Fig. 3. Results from the lexicon algorithm. 49.1% of the textual contents are neutral, 8.8% are negative and 42.1% are positive.

B. Test and Results of the Second System

In this experiment, fifty seven preprocessed textual contents, collected from Twitter have been parsed in the system. These contents are then categorized as negative, neutral and positive (Fig. 5) by the lexicon algorithm. The environment details under which the textual contents have been made are parsed in the sentiment prediction algorithm to predict the sentiment of the contents that would be framed under those specific environments.

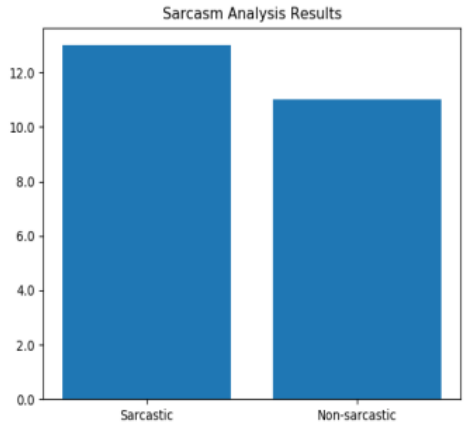

Fig. 4. Results from the sarcasm analysis algorithm. 13 textual contents of the positive text contents are sarcastic and 11 are non-sarcastic.

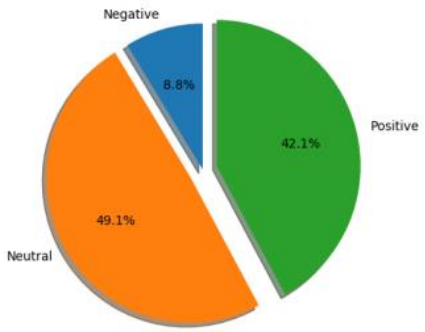

Fig. 5. Results from the lexicon algorithm. 49.1% of the textual contents are neutral, 8.8% is negative and 42.1% is positive.

If the result from the lexicon algorithm and the result from the sentiment prediction algorithm for a content, are similar, this later is categorized as non-sarcastic else, if the results are not similar, the content is categorized as sarcastic. Fig. 6 shows the results from the sentiment prediction algorithm.

For each comment, the result of the sentiment prediction algorithm has been compared with result of the lexicon algorithm. In case these results are similar, the content is considered as non-sarcastic else it is considered as sarcastic. The environment details used in this experiment are assumptions that have been made since the required data were not

available and the proposed system was an innovative one at the time of experiment.

V. CONCLUSION

The aim of this study was to propose ways to extend the lexicon algorithm in order to build systems that would be more efficient for sarcasm detection. This aim had been successfully met as two systems have been developed to address this situation. However, in the first system, it had been noticed that the training set of the sarcasm analysis algorithm must be relevant to the actual data that need to be analyzed in order to obtain meaningful results and to improve the accuracy of the system. The second system constitutes a vast area of study. Some work need to be done in order to develop a system that would allow the collection of environment details under which the textual contents would be made on social media platforms. A consolidated way of computing the sentiment polarity of the environments based on their details should also be developed.

REFERENCES

[1]. Cambridge University Press, 2018. sarcasm. [Online] Available at: https://dictionary.cambridge.org/dictionary/english/sarc asm [Accessed 20 January 2018].

[2]. Palanisamy, P., Yadav, V., & Elchuri, H. (2013). Serendio: Simple and Practical lexicon based approach to Sentiment. 543-548

[3]. Jurek, A., Mulvenna, M. D., & Bi, Y. (2015). Improved lexicon-based sentiment analysis for social media analytics. SpringerOpen, 4-9.

[4]. Kiilu, K. K., Okeyo, G., Rimiru, R., & Ogada, K. (2018). Using Naïve Bayes Algorithm in detection of Hate Tweets. International Journal of Scientific and Research Publications, 99-107.

[5]. Rathan, K., & Suchithra, R. (2017). Sarcasm detection using combinational. Imperial Journal of Interdisciplinary Research, 546-551

[6]. Sathya, R., & Abraham, A. (2013). Comparison of Supervised and Unsupervised. International Journal of Advanced Research in Artificial Intelligence, 34-38.

[7]. Dataquest, 2018. Top 10 Machine Learning Algorithms for Beginners. [Online] Available at: https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/ [Accessed 15 September 2018].

[8]. Haripriya, V., & Patil, D. P. (2017). A Survey of Sarcasm Detection in Social Media . International Journal for Research in Applied Science & Engineering Technology, 1748-1753.

[9]. Musto, C., Semeraro, G., & Polignano, M. (n.d.). A comparison of Lexicon-based approaches for Sentiment Analysis of microblog posts.

[10]. Saxena, R., 2017. How the naive bayes classifier works in machine learning. [Online] Available at: http://dataaspirant.com/2017/02/06/naivebayes-classifier-machine-learning/ [Accessed 10 February 2019].

[11]. Gandhi, R., 2018. Naive Bayes Classifier. [Online] Available at: https://towardsdatascience.com/naivebayes-classifier-81d512f50a7c [Accessed 10 February 2019].

[12]. RAY, S., 2017. 6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R). [Online] Available at: https://www.analyticsvidhya.com/blog/2017/09/naivebayes-explained/ [Accessed 10 February 2019].

[13]. Aggarwal, S., & Kaur, D. (2013). Naïve Bayes Classifier with Various Smoothing. International Journal of Computer Trends and Technology, 873-876