TECHNICAL JOURNAL Vol 1, No.1, July 2019
Nepal Engineers' Association, Gandaki Province
ISSN : 2676-1416 (Print)
Pp.: 1-9

A HYBRID APPROACH FOR SARCASM DETECTION

S. Luintel, R.K. Sah, B.R. Lamichhane
Department of Electronics and Computer, Paschimanchal Campus
Lamachaur, Pokhara, P.O. Box : 46, Fax No. : 061-440158, NEPAL
(e-mail: sadhanaluitel@gmail.com)

ABSTRACT

There is an excessive growth in user generated textual data due to increment in internet and social media users which includes enormous amount of sarcastic words, emoji, sentences. Sarcasm is a nuanced form of communication where individual states opposite of what is implied which is done in order to insult someone, to show irritation, or to be funny. Sarcasm is considered as one of the most difficult problems in sentiment analysis due to its ambiguous nature. Recognizing sarcasm in the texts can promote many sentiment analysis and text summarization applications. So for addressing the problem of sarcasm many steps have been adopted for sarcasm detection. Different preprocessing techniques such as Hypertext markup language removal, stop words removal, etc. have been done. Similarly, conversion of the emoji and smileys into their textual equivalent has been performed. Most frequent features has been selected and a hybrid cascade and hybrid weighted average approaches which are the combinations of the algorithms random forest, naïve Bayes and support vector machine have been used for sarcasm detection. The comparison of these two approaches on different basis has been done which has shown cascade outperformed weighted approach. Moreover, comparison of cascade approaches in terms of the algorithm placement has also been performed in which random forest has proved to be the best.

KEYWORDS: *cascade, detection, hybrid, sarcasm, sentiment analysis*

INTRODUCTION

People express their opinion on social media site or e-commerce sites these days due to advancement of internet facility and technological media. Lots of textual data is generated because of this excessive growth in the use of social media platforms which consists of all types of opinions. It includes opinions which reflects ones thinking about the product or current affairs. The definition of sarcasm can be done as use of words that means the opposite of what the expresser really wants to say in order to insult someone, to show irritation or to be funny (Sreelakshmi and Rafeeque, 2018; Gidhe and Ragha, 2017). Sarcasm is one of the major linguistic concept used in social media in present context. While speaking, it is very easy to distinguish sarcasm utilizing pitch of voice, gesture, facial expression etc. But in textual data, it is difficult to detect sarcasm due to lack

of described factors. For example, "Wow, there is huge amount discount for buying television." This sentence considered can be taken as a compliment. However, considering following sentence: "Wow, there is huge amount of discount while buying television but I don't want to buy any of the television." This sentence clarifies that person did not mean what he/she said. Hence, it becomes a difficult job for a normal person to detect what the person wanted to express (Chaudhari and Chandankhede, 2017).

There are many fields such as natural language processing, sentiment analysis, opinion mining for which data in textual form is preliminary unit of analysis. Since the textual dataset has sarcastic content of some form in it, the sarcastic contents can convert the polarity of the data into opposite of what is meant. So, if the impact of sarcasm in sentiment analysis or other fields is ignored then the polarity of sentence or overall content may become diverted and a high level of accuracy and reliability is not achieved. So, it is important to detect sarcasm for accurate sentiment analysis, review processing and natural language processing. And hence the need of the sarcasm detection system arises (Parmar et al., 2018; Lunando and Purwarianti, 2018).

Rule based and machine based approaches have widely been used for sarcasm detection process. Cascading of algorithm (Hybrid model) has been performed to solve problem faced in intrusion detection system. Naïve Bayesian (NB) and Support Vector Machine (SVM) have been combined to maximize accuracy, advantages of both approaches have been integrated for metrics maximization (Sagale and Kale, 2014). Feature selection scheme has been performed by using Hidden Markov Model -Latent Dirichlet Allocation (HMM-LDA). Depending upon the selected features sentiment classification has been performed using hybrid Naïve Bayes and SVM approach (Sumanthi and Sheela, 2015). A weighted hybrid model utilizing Support Vector Machine and Naïve Bayes for anomaly discovery has been used. K-fold cross validation has been utilized to figure the error which has concluded that hybrid approach is best (Shakya and Sigdel, 2017)

In this paper, a hybrid algorithm approach has been used to detect whether a given sentence, comment or paragraph is sarcastic or non-sarcastic and the approach used is compared with weighted average approach.

EXPERIMENTAL

Figure 1 represents the overall workflow of the proposed sarcasm detection system. The sarcasm detection system has five major units:

Dataset collection

The data has been taken from Github repository. The data of textual form has sentences with slangs, emoticons, smileys and so on. The training dataset has columns of Id, sentiment and review.

Pre-Processing

The first task is preparing the corpora for the application of algorithm. Several preprocessing techniques need to be applied to clean sentiments. This involves activities such as

denoising text, removing hypertext markup language (HTML) markup, removing between square brackets, expanding contractions and tokenization.

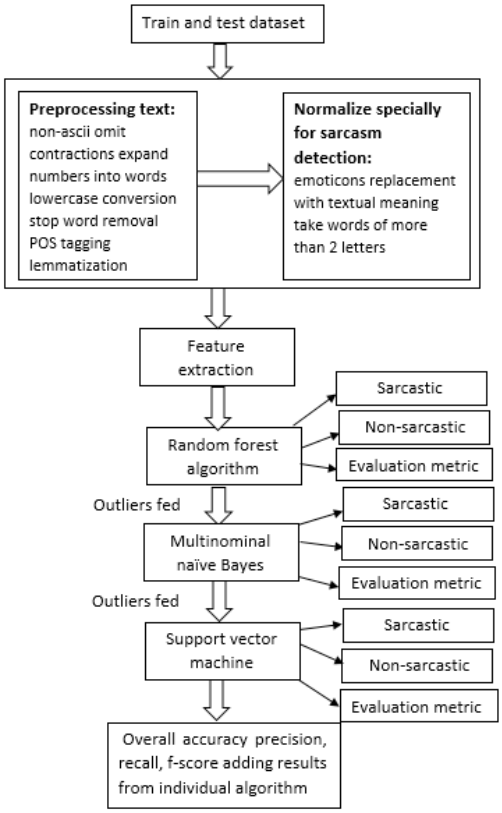

Figure 1. Proposed Sarcasm Detection System the execution of code.

Normalization

Normalization puts all words on equal footing, and allows processing to proceed uniformly. This includes processes such as removal of stop words, lemmatization, part-of-speech tagging etc.

Feature Extraction

It is the extraction of specific features which can represent what the reviewer is trying to say into numerical features. It is not computationally feasible to take all the features into consideration as there are constraints like time and speed involved in

Training

The dataset consisted of 54206 training dataset and 3742 testing data set. Also there was a test dataset with true sentiments. Only 15002 training set and 3742 test dataset was taken as dataset in lot 1 because of computational complexity. The training dataset was also partitioned into three datasets of almost 5000 dataset each for lot 2 with test dataset being same. Pre-processing, normalization and feature extraction were performed in the dataset and then they were fed to the classifiers. The proposed system has been modeled using three classifiers namely Support Vector Machines (SVM), Random Forest and naïve Bayes algorithm.

Approaches

Cascade Approach

The preprocessed dataset is fed into the algorithms one after another.

First Lot

The training set of each algorithm remains the same which is 15002.

Second lot

The training set of each algorithm remains different. They have been divided into groups of 5000 dataset each. Each algorithm has new dataset to train from.

The test dataset is 3742 for the first algorithm only in both the lots above. Then the test dataset is finally classified into:

1) Sarcastic:1
2) Non-sarcastic:0

The concept of hybrid approach comes into play when output of first algorithm is fed again for classification into second algorithm. This is done by feeding into the second algorithm the incorrectly calculated test dataset of the first algorithm. So for the second algorithm the test dataset is reduced. It classifies the test dataset into sarcastic and non-sarcastic. The same process is repeated for the third algorithm as well. And the incorrectly fed output is fed for prediction for the third algorithm. The test dataset is again reduced. This algorithm again classifies the test dataset into sarcastic and non-sarcastic. Then the final output is expected to bring a boost in accuracy than the previously applied algorithms at first and second.

Weighted Average Approach

The preprocessed dataset is fed into the algorithms one after another.

First Lot

The training set of each algorithm remains the same which is 15002.

Second lot

The training set of each algorithm remains different. They have been divided into groups of 5000 dataset each. Each algorithm has new dataset to train from.

The test dataset is 3742 for the all algorithms and both the lots above. Then the test dataset is finally classified into following two categories by all algorithms individually.

1) Sarcastic:1
2) Non-sarcastic:0

Then the root mean square error (RMSE) of each algorithms is computed by formula:

$$\text{RMSE}=\sqrt{\sum_{i=1}^{n}\left(\frac{1}{\text{n}}\right)*\left(\text{true labels - predicted labels}\right)^2} \quad \ldots\ldots (1)$$

In the above equation (1), n is total number of test data. True labels are the correct labels of the test data and predicted labels are the output labels of algorithms. The predicted labels are replaced according to the algorithms applied. So we have three root mean square errors which are:

1) $\text{RMSE}_{\text{rf,}}$
2) $\text{RMSE}_{\text{nb,}}$
3) RMSE_{svm}

The weight weight_{j} of j^{th} member algorithm is computed using error as:

$$\text{weight}_{\text{j}} = \frac{1}{\varepsilon^{j}} \times \sum_{i=1}^{M} \frac{1}{\varepsilon^{i}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The equation (2) demonstrates that a bigger weight is allocated to a candidate algorithm with higher accuracy.

Now weight for all the algorithms is calculated as:

$$\text{weight}_{\text{algorithms}} = \frac{\text{RMSE}_{\text{algorithm}}}{\text{RMSE}_{\text{algorithm1}}+\text{RMSE}_{\text{algorithm2}}+\text{RMSE}_{\text{algorithm3}}} \quad .. (3)$$

The weight and RMSE have an inverse relation. The algorithm with highest accuracy has lowest root mean square error and its weight is highest. So during weight assigning algorithm that has highest accuracy gets highest weight. So the weight are computed as that and the weight of algorithms are:

1) $weight_{rf}$
2) $weight_{nb}$
3) $weight_{svm}$

The weights chosen has the constraint defined by $weight_{rf}+weight_{nb}+weight_{svm}=1$ ……. (4)

Equation (4) means that weight of all algorithm added should equal 1.

Finally, the prediction of hybrid algorithm is given by $y_h(x)= \sum_{i=1}^{N} weight_i \times y_i(x)$ …….... (5)

$y_i(x)$ in equation (5) refers to the measures like accuracy, precision, f-score etc.

Hence, in this way weighted average calculation is implemented.

RESULTS AND DISCUSSIONS

Output

The output obtained after the application of different algorithms in the train and test dataset is in the form of 0 and 1 for the test dataset. 0 is supposed to represent the sentiments with sarcastic orientation and 1 is supposed to represent sentiments with non-sarcastic orientation.

Validation

There is test dataset containing of 3742 data with correct labels.

1) Sarcastic: 1
2) Non-sarcastic: 0

The classified test set is tallied with this dataset for validation.

Comparison between two approaches

First lot: There were 15002 train dataset and 3742 test data for which following observation were obtained for first batch.

Second lot: For this lot first the training dataset (15002) were divided into 3 groups of almost 5000 dataset each and then those were fitted as training dataset to three different algorithms. So for each time the test dataset learned from completely unseen training datasets. This step has been taken so that the over fitting problem is reduced to some extent.

Table 1 below shows the comparison among hybrid approaches in which the time to obtain output from weighted average is 150.66 seconds and from cascade approach is 142.94 seconds, so weighted average is comparatively slower here. Accuracy comparison tells that the cascade approach is quite accurate than weighted average where the accuracy for the weighted approach is 67.29% and for cascade approach is 90.37%. The precision and recall for weighted average are 55.97% and 66.99% whereas for cascade approach they are 74.7% and 98.41% respectively. It can be observed that cascade approach is more precise compared to weighted approach and its recall is also extremely ahead compared to other. The F-score is found to be 60.82% for weighted average and 82.19% for cascade approach, in this too cascade approach is ahead. So after all the comparisons made for this lot, the cascade approach is ahead of weighted average approach.

Criteria	Weighted average			Cascade approach		
	Random	Naive	SVM	Random	Naive	SVM
Time (algorithms only)	150.66 seconds			142.94 seconds		
Accuracy	70.06	66.32	66.46	69.61	13.04	7.72
(%)	67.29			90.37		

Precision	58.57	55.50	54.40	58.17	11.01	5.83
(%)	55.97			74.7		
Recall	71.95	56.52	71.31	71.31	9.37	16.47
(%)	66.99			98.41		
F-score	64.57	56.05	61.72	64.01	10.47	7.85
(%)	60.82			82.19		

Table 1 Comparison between two hybrid approaches (Lot 1)

Criteria	Weighted average			Cascade approach		
	Random	Naive	SVM	Random	Naive	SVM
Time (algorithms only)	24.08 seconds			24.75 seconds		
Accuracy	66.75	66.46	62.25	66.91	16.99	8.89
(%)	65.26			92.81		
Precision	53.31	57.69	50.45	55.98	16.52	4.57
(%)	58.48			74.88		
Recall	64.2	43.34	66.59	64.97	10.71	12.6
(%)	54.46			91.59		
F-score	59.42	49.99	57.41	60.14	13.43	6.30
(%)	55.64			82.64		

Table 2 Comparison between two hybrid approaches (Lot 2)

Table 2 above is the comparison between cascade and hybrid approaches in lot 2 where it can be seen that the time consumed to obtain output from weighted average is 24.08 seconds and from cascade approach is 24.57 seconds. This shows cascade approach is slightly slower here. Accuracy comparison tells that cascade approach is quite accurate where the accuracy for the weighted approach is 65.26% and for cascade approach is 92.81%. The precision and recall for weighted average approach are 58.48% and 54.64% and for cascade approach they are 74.88% and 91.59% respectively. It can be observed that cascade approach is precise and its recall is extremely ahead compared to other. The F-score is found to be 55.64% for weighted average and 82.64% for cascade approach, in this too cascade approach is ahead. So after all the comparisons made for this lot, the cascade approach is ahead of weighted average approach. Hence, after observing results obtained from all ratio cascade approach is found to be more efficient than weighted average approach.

Representation of evaluation metrics with bar graph

Figure 2 below shows the precision of weighted average approach and cascade approach in the lot1 and lot 2. The precision for weighted average for lot 1 and lot 2 are 55.97% and 58.48% whereas for cascade approach, the precision scores are 74.75% and 74.78%. It can be observed that cascade approach is precise in both batches.

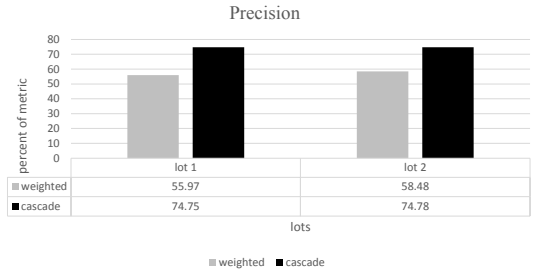

Figure 2. Precision of both approaches in lot 1 and lot 2

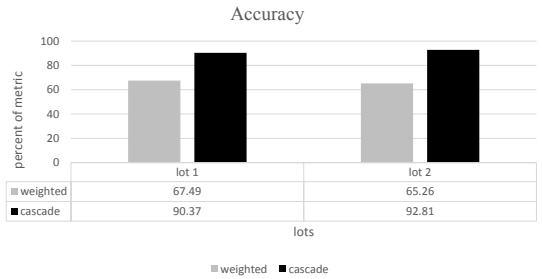

Figure 3. Accuracy of both approaches in lot 1 and lot 2

Figure 3 above shows the representation of the accuracy in percentage of weighted average approach and cascade approach in the lot1 and lot 2. The accuracy for lot 1 and 2 is found to be 67.49% and 65.264% for weighted average and 90.37% and 92.81% for cascade approach in lot 1 and lot 2 respectively. The accuracy of cascade is more in both lots.

Comparison of statistics between three cascade approaches

For the comparison, the following training and test ratio was taken into consideration.

Training dataset: 15000 (split randomly into 5000 dataset each and fed to different algorithms) and

Test dataset: 3742

The comparison of the statistics portion has been employed particularly for the case of cascade approach. It has been conducted to see whether the placement of the algorithm in the code has any change in the accuracy and other metrics.

1) First approach: It has SVM at first then naïve Bayes and random forest.
2) Second approach: It has naïve Bayes at first then SVM and random forest.
3) Third approach: It has random forest at first then naïve Bayes and SVM.

Table 3 below shows the comparison made among three cascade approaches on the basis of placement of the algorithm in the code in which the time to obtain output from first approach is 24.49 seconds, second approach is 25.001 seconds and for the third approach is 24.57 seconds. So approach with SVM first is faster among three approaches. Accuracy comparison among the approaches tells first approach has accuracy of 89.68%, second approach has 90.05% accuracy and third approach has 92.81% accuracy. Here, it can be seen that accuracy of random forest is best in comparison of three approaches. The precision and recall for first approach are 70.22% and 86.33% whereas for second they are 85.71% and 85.16% and for the third approach are 74.88% and 91.59% respectively. It can be observed that second approach is more precise whereas in case of recall the third approach is quite ahead of both the approaches. The F-score is found to be 76.28% for first approach and 85.43% for second approach and for third approach 82.64%.

Hence, after observing results from two approaches it can be concluded that placement of algorithm plays a role in the accuracy and other metrics like precision, recall, etc. It shows that whenever the algorithm with higher accuracy is placed at first the other metrics corresponding to it are better as compared to when they are placed at some other position in code in majority of the cases.

Table 3 Comparison between cascade approaches (algorithm placement)

Criteria	(Naïve Bayes first)			(SVM first)			(Random first)		
	Nb	SVM	Ran	SVM	Nb	Ran	Ran	SVM	Nb
Time (algorithms)	25.001 seconds			24.49 seconds			24.57 seconds		
Accuracy (%)	66.46	15.47	8.12	63.61	17.07	8.09	66.91	16.99	8.89
	90.05			89.68			92.81		
Precision (%)	57.69	24.84	3.17	51.61	12.17	6.47	55.98	16.52	4.57
	85.71			70.22			74.88		
Recall (%)	43.34	30.37	11.45	62.25	8.17	12.8	64.97	10.71	12.6
	85.16			86.33			91.59		
F-score (%)	49.49	28.37	5.68	57.64	10.30	8.34	60.14	13.43	6.30
	85.43			76.28			82.64		

Representation of evaluation metrics of cascade approaches

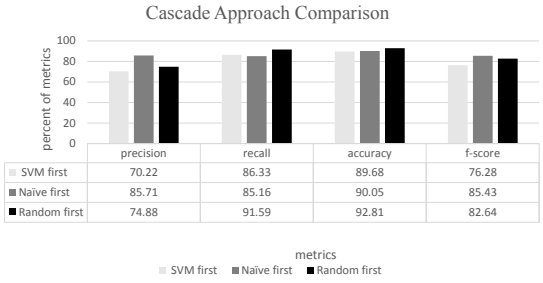

Figure 4. Comparison of metrics among three cascade approaches

Figure 4 above shows the comparison among three cascade approaches on the basis of the placement of algorithm. Precision comparison tells that approach with naïve Bayes at first is ahead of all. Recall comparison displays that approach third is best among all. The comparison of approaches on the basis of f-score tells that naïve Bayes approach when placed at first outperforms other three approaches. Hence from the comparison of statistics it can be told that approach with random forest at first performs better in majority of criteria.

CONCLUSIONS

Accuracy has been evaluated as the key method for efficiency of algorithms which concluded that cascade approach is more accurate than weighted average approach where accuracy of cascade was 90.37% and weighted was 67.29% (lot 2). Other measures like f-score, recall and precision also concluded that cascade approach is more efficient. Also there was comparison of approaches on the basis of algorithm placement in which random approach placed at first in the code outperformed others with accuracy of 90.37% (lot 2).

As for further improvements, the sentiments categorized into sarcastic and non-sarcastic can further be classified into positive sarcastic and negative sarcastic. Others algorithms such as linear regression, neural network, genetic algorithm, etc. can be employed for the classification purpose. Future work may center on covering the different form of sarcasm in sarcasm detection approach and to detect sarcasm in new languages.

ACKNOWLEDGEMENT

The authors thank to Prof. Dr. Subarna Shakya, Prof. Dr. Nanda Bikram Adhikari, Sitaram Pokherel, Hari Baral, Hari K.C, Sharan Thapa and Ramesh Thapa for their insightful comments and encouragement. Their contributions are sincerely appreciated and gratefully acknowledged.

REFERENCES

Chaudhari P. and Chandankhede C. (2017). Literature survey of sarcasm detection. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2041-2046. doi: 10.1109/WiSPNET.2017.8300120

Gidhe P. and Ragha L. (2017). Sarcasm detection of non # tagged statements using MLP-BP. *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 1-4. doi: 10.1109/ICAC3.2017.8318756

Lunando E. and Purwarianti A. (2013). Indonesian social media sentiment analysis with sarcasm detection. *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 195-198. doi: 10.1109/ICACSIS.2013.6761575

Parmar K., Limbasiya N. and Dhamecha M. (2018). Feature based Composite Approach for Sarcasm Detection using MapReduce. *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*, 587-591. doi: 10.1109/ICCMC.2018.8488096

Sagale Amit D. and Kale Swati G. (2014). Combining Naive Bayesian and Support Vector Machine for Intrusion Detection System. *2014 International Journal of Computing and Technology*, *1*(3), 2348 – 6090.

Shakya S. and Sigdel S. (2017). An approach to develop a hybrid algorithm based on support vector machine and Naive Bayes for anomaly detection," *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 323-327. doi: 10.1109/CCAA.2017.8229836

Sreelakshmi K. and Rafeeque P. C. (2018). An Effective Approach for Detection of Sarcasm in Tweets. *2018 International CET Conference on Control, Communication, and Computing (IC4)*, 377-382. doi: 10.1109/CETIC4.2018.8531044

Sumathi N. and Sheela Dr. T. (2017). An Efficient Sentiment Analysis by using Hybrid Naive Bayes and Svm Approach in Banking Institutions. *2017 International International Journal of Civil Engineering and Technology (IJCIET)*, *8*(12), 373–391.