

Sarcasm Detection using Deep Learning

Seema Kedar¹, Keshav Bhardwaj², Nishant Igave³, Sagar Jagtap⁴, Shripriya Khadakkar⁵, Sahil Bhirud⁶

¹JSPM's RSCOE, Pune, India,

²TCS, Pune, India,

³Corecomplete, Pune, India

⁴TCS, Pune, India

⁵Fiserv, Pune, India

⁶MS, The Graduate School UNC Charlotte, US

Article History: Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27 January 2021; Published online: 05 April 2021

Abstract: It is becoming a trend in current society to use complex and indirect statements for communication which includes metaphorical language, proverbs and other similar forms. One of such communication form is sarcasm. Sarcastic statements can have symbolized, hidden or even entirely opposite meaning from the conveyed statement. Sarcasm is inherently ambiguous in nature which makes it very difficult to understand even for humans let alone machines. In this paper, we have implemented sarcasm detection based upon difference and similarity between facial emotion of the person and sentiment of his verbally conveyed message.

Keywords: ngram, CNN, logistic regression, sarcasm detection

1. Introduction

Sarcasm is a “sharp, bitter, or cutting expression or remark; a bitter gibe or taunt” that mean the opposite of what they say, made to criticize someone or something in a way that is amusing to others but annoying to the person criticized. It might employ ambivalence, although sarcasm is not necessarily ironic. Most noticeable in spoken word, sarcasm is mainly distinguished by the inflection with which it is spoken and is largely context-dependent.

Sarcasm raises the complexity level of any communication and necessitates a higher level of intelligence to understand the true meaning of the statement behind its literal meaning. Automatic sarcasm detection is a system that recognizes sarcastic features in face-to-machine conversation. A person being sarcastic will have difference in his/her facial emotions and verbal emotions. This work is to build precise real time system that will analyses the correlation between facial and verbal emotions and the flow of change of facial emotions during and after a verbal statement. Result thus obtained will be analyzed to detect sarcasm in face-to-machine conversation. Facial emotion will be obtained by real time face recognition and emotion analysis system. And verbal emotions will be obtained by text extraction from voice and applying sentiment analysis over text.

In this paper, Sarcasm is detected by correlating emotions from text and face. Sentiment analysis from text has many approaches. Haseena Rahmath, Tanvir Ahmad [4] discovers various techniques for sentiment analysis from text given as: machine learning based, lexicon based and hybrid technique. Machine learning based technique belongs to supervised classification sentiment analysis. Two sets of data are required: training and test data. Training dataset is used by classifier for building of the model for sentiment analysis and test data is used to check whether module built by classifier performs well or not. The most commonly used features are as follows: term presence and their frequency, part of speech information, negations, opinions and word phrases. Lexicon based technique contains list of words use to express peoples emotion. There are three methods for construction of sentiment lexicon: corpus-based, manual construction and dictionary-based methods. Hybrid technique is a combination of machine-based and lexicon-based method. Hybrid method improves the accuracy and performance of sentiment analysis from text. Machine-based technique shows relatively better performance than unsupervised lexicon-based, but lexicon-based is also important due to large amount unlabeled everyday data. Support Vector Machine (SVM) reports higher accuracy than other algorithms. Hybrid approach achieves greater accuracy from supervised machine-based technique and stability from unsupervised lexicon-based approach.

Carlos Busso, et al[5] analyzed the strengths and weaknesses of facial expression classifiers and acoustic emotion classifiers. In this paper, they have compared many emotion recognition systems to see which system performs better. From which emotion recognition by facial expression yields more than any other emotion recognition system. The muscles of the face can be changed and the tone and the energy in the production of the speech can be intentionally modified to communicate different feelings which make it difficult to detect emotion from facial expressions or tone. To detect facial expressions the features used are based on local spatial position or displacement of specific points and regions of the face. They proposed an emotion recognition system, in

which the optical flow is used to extract the muscle movements which are done with 11 windows which are manually located in the face. To classify into different emotions K-Nearest Neighbor algorithm is used. The accuracy for classification is 80% with four emotions: happiness, anger, disgust and surprise. A similar method is proposed in which instead of using facial muscle actions, they have built a dictionary to convert motions associated with edge of the mouth, eyes and eyebrows, into a linguistic, per frame, mid-level representation. With the use of this dictionary approach they achieved 88% accuracy in classification of six basic emotions.

2. Literature Survey

Aditya Joshi, et. al. [1] describes datasets, approaches, trends and issues in sarcasm detection and discussed representative performance values. Sarcasm has a negative implied sentiment, but may not have a negative surface sentiment. Data is divided into four categories: short text (tweets on Twitter), long text (discussion forum posts), transcripts (transcripts of a TV show), and other miscellaneous datasets. Rule-based approaches attempt to identify sarcasm through specific evidence. The evidence in rule-based approach is obtained in the form of rules that rely on indicators of sarcasm. Semi-supervised pattern extraction is used to identify implied sentiment. Hash tag tokenizer is used to divide hash tags made of concatenated words. Context beyond target text is one of the milestones observed in sarcasm detection research.

Rachel Rakov and Andrew Rosenberg [2] identify features that may indicate sarcasm. These features include mean of feature, standard deviation of feature, feature range, mean amplitude, amplitude range, speech rate, harmonics-to-noise ratio. From the given features, reduction in mean of feature, decrease in variation in feature and change in harmonics-to- noise ratio were found to be indicative of sarcastic speech. Common patterns of pitch and intensity contours are found using k-mean clustering algorithm. Prosodic sequences of pitch and intensity contours using k-mean centroids for training of ngram model for detection of sarcasm. Using a Simple Logistic (LogitBoost) classifier predicts sarcasm with 81.57% accuracy. Performance metrics for the system is exact match accuracy and dice score.

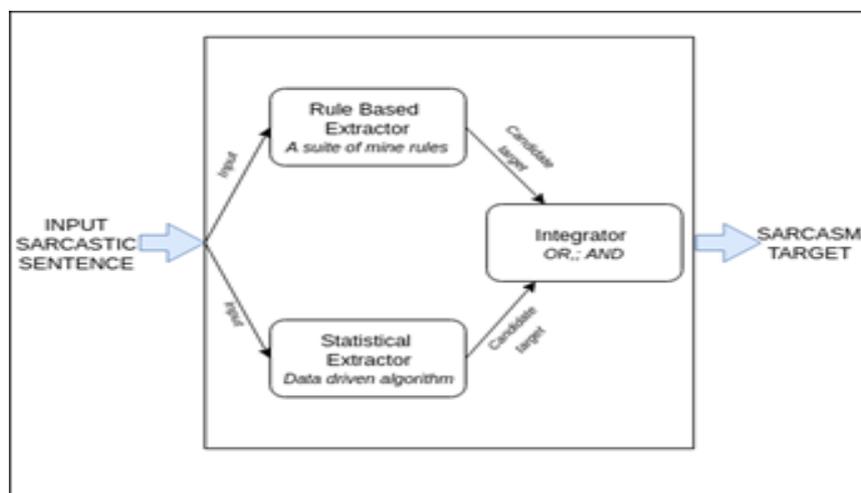


Figure 1. Sarcasm Target Identification

Pranav Goel, et. al. [3] proposed sarcasm detection system based on combination of rule and statistical classifier as shown in Figure 2. A new dataset is introduced for sarcasm detection, manually annotated for the sarcasm target in snippets and tweets based on formulation of task. Rule based extractor includes nine rules. It takes text sarcastic input and obtains a set of candidate sarcasm targets. For generation of candidate set of sarcasm targets, a weighted majority approach is used. Statistical based extractor takes input a word along with its features and returns if the word is a sarcasm target. Integrator induces the sarcasm target from output of the two extractors. Two configurations of the integrator are considered: hybrid OR and hybrid AND.

From the previous research it can be seen that, [1] Sarcasm can be detected from the text which are embedded in the text. For example, Man is as useful to woman as cycle is to a fish. In the given text sarcasm is embedded in the sentence itself, it can be detected with efficiency. It is difficult for sentiment analysis from text which needs context of the situation. For Example, Your plan sounds awesome. This statement can be identified as non-sarcastic if the context of the situation is not known or understood. [1] Sarcasm is detected from amplitude; pitch, that is, how the tone of the person changes when he/she speaks sarcastic statement. But sarcasm cannot be detected efficiently if person does not change the tone of speech while speaking. For this proposed system, we

have implemented different modules for detection of Sarcasm. First module detects emotions from the text. Facial Emotion recognition detects emotion from user's facial expressions. Third module detects sarcasm by correlating both the modules.

3. Proposed System

People use sarcasm in order to criticize other people or to make them silly. This can happen with machine also. Sometimes user can say something very opposite than the thing which he wants to get done from machine (in sarcastic way), then machine has to understand sarcasm from users facial expressions or by the pitch of persons tone or else machine will do that work and will make user more upset. So, we are developing a system which will help machine to understand the sarcasm by user's facial expression or the text given as input.

There will be two types of input for the system:

1. Visual Inputs: In this type of input the machine will capture the images of user's facial expressions. And according to the images captured the machine will firstly recognize the mood of user by using deep learning. This will help to detect sarcasm.
2. Text Inputs: In this type of input the machine will detect emotion from the given input text using deep learning. And according to users mood machine will be able to detect sarcasm. The system will correlate the both input and from that input the system will detect sarcasm more accurately.

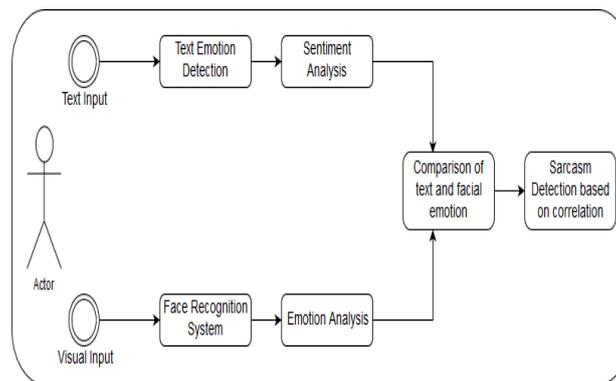


Figure 2. Automatic sarcasm detection architecture

4. Implementation

Use either Implementation for the proposed automatic sarcasm detection system is divided into three modules:

Sentiment analysis from text
 Facial Emotion Recognition
 Sarcasm Detection

1. Sentiment analysis from text

For the purpose of sentiment analysis from text, supervised hashtag dataset from twitter is used. Data collected is labeled with positive and negative tweets. Positive tweet is represented as value one. Negative tweet is represented as value zero. For example on e-commerce website review Product is great. I am using since 6 months and still in good condition is positive statement labeled with value one. And review Product is fake is negative statement labeled as zero.

First step is data pre-processing, insignificant characters which do not contribute for analysis and emotion detection are removed. This insignificant character includes hashtag, URLs, @ symbols, username and special characters. Symbols such as! (Exclamatory punctuation) which contributes to emotion of the text is replaced by specific words. Replacing emoji with respective words using Unicode Emoji Charts file. In negation handling, don't is replaced by do not, can't is replaced by cannot and so on).All words are converted to lower alphabetical order. Stemming and lemmatization are applied on the words, that is, for stemming word is stripped to stemmed word for example, cookery to cookerie and for lemmatization word is written in its root word, for example, cooking to cook. Data is divided into training, dev and test set. The ratio to split data is 98/1/1, 98% of data as the training set, and 1% for the dev set (for cross validation), and the final 1% for the test set. The dataset has

more than 1.5 million entries. Only 1% of the whole data gives more than 15,000 entries. Train set has total 1564098 entries with 50.00% negative, 50.00% positive. Validation set has total 15960 entries with 50.40% negative, 49.60% positive. Test set has total 15961 entries with 50.26% negative, 49.74% positive. Feature extraction is to use text inputs for machine learning; the given text is needed to be converted into numerical format. There are few methodologies for implementing the same as using a corpus of words where the context is not considered but frequency of the words is the only contributing factor, another method is word to vector where each word is converted to a specific vector here vector value of similar words are close to each other while words not having any similarity will have larger differences in their vector values. Another representation that is being used in current approach count vectorize. In count vectorize method vectors of frequency of words are created. Suppose one is having following three documents in a corpus, “I’m happy to see you”, “I’m there for you”, “you work hard for you to win”.

Table 1. Count Vectorization

	I'm	happy	to	see	you	there	for	work	hard	win
doc1	1	1	1	1	1	0	0	0	0	0
doc2	1	0	0	0	1	1	1	0	0	0
doc3	0	0	1	0	2	0	1	1	1	1

Length of vector for each document will always be same as all distinct words in specific order are used for generation of vector. Frequency of a word in a document is use as the numerical value part of the vector of respective location as it is evident that for document 3 frequency of occurrence for word ‘happy’ is zero but for word ‘you’ it is two and in the similar manner vectors for all the documents are created.

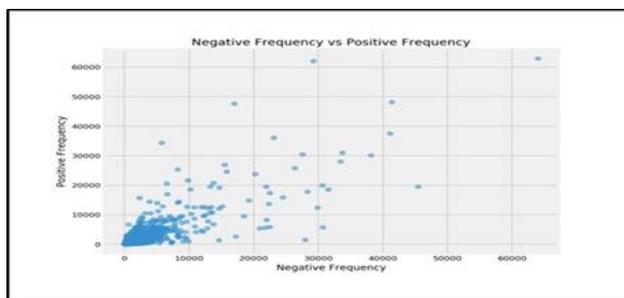


Figure 3. Word Frequency Comparison

The Figure 3 displays word frequency comparison. Each dot represents a word in the collection which could be used as a feature, x and y-axis represents frequency of that word in negative and positive tweets respectively. Assume a line dividing this plane at 45 degree from the point of origin to infinity.

Now all the points or words closer to the line are considered less important features than the points that are further away from the line because points closer to the line are contributing less in deciding the sentiment of the statement which is evident as their occurrence in positive statements and in negative statements is similar.

Ngram is takes n words as a feature. As the n increases in ngram, accuracy of model built increases till certain point. For example, “I am not very happy”. For unigram, single word is considered as feature. (I), (am), (not), (very), (happy). In bigram two words are considered as feature. (I, am), (am, not), (not, very), (very, happy). In trigram, (I, am, not), (am, not, very), (not, very, happy).

The best validation set accuracy for each n-gram is given in Figure 4. Unigram: 80,000 and 90,000 features at validation accuracy 80.28%. Bigram: 70,000 features at validation accuracy 82.25%. Trigram: 80,000 features at validation accuracy 82.44% Logistic Regression is applied on the data with ngram as feature.

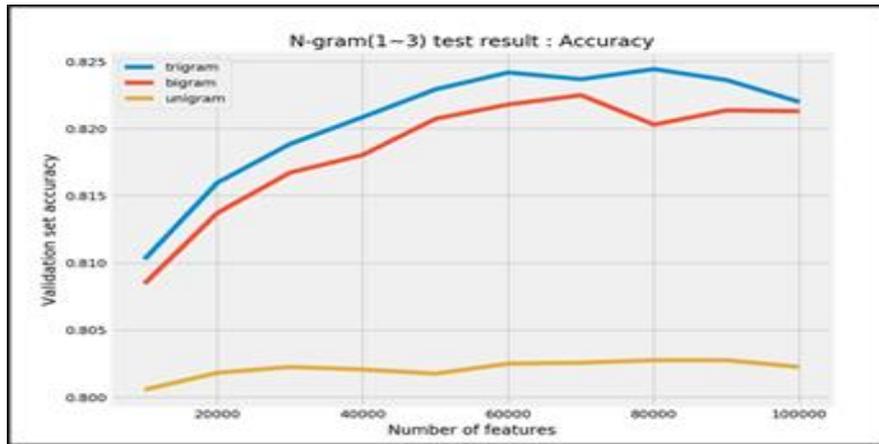


Figure 4. Performance comparison of unigram, bigram and trigram

2. Facial Emotion Recognition

The dataset used for facial emotion recognition is fer2013 dataset published on International Conference on Machine Learning (ICML). For facial emotion recognition, dataset consists of 35,887 grayscale, 48x48 sized face images with various seven emotions. Emotion labels in the dataset are given as in table II.

Table 2. Facial Emotion Dataset

Index	Images	Emotion
0	4593	Angry
1	547	Disgust
2	5121	Fear
3	8959	Happy
4	6077	Sad
5	4002	Surprise
6	6198	Neutral

The fer2013 data is loaded and generated to scaled image. This generated scaled images and their labels are loaded. Loaded images are reshaped to 48x48 images.

CNN algorithm is used for emotion recognition from facial features. CNN (Convolutionary Neural Network) is class of deep neural networks, most commonly applied to analyze the visual imagery. We are applying sequential model with two layers of CNN.

Two layers in CNN used are as follows:

- (a). The first layer will have 32-3x3 filters.
- (b). The second layer will have 64-3x3 filters.

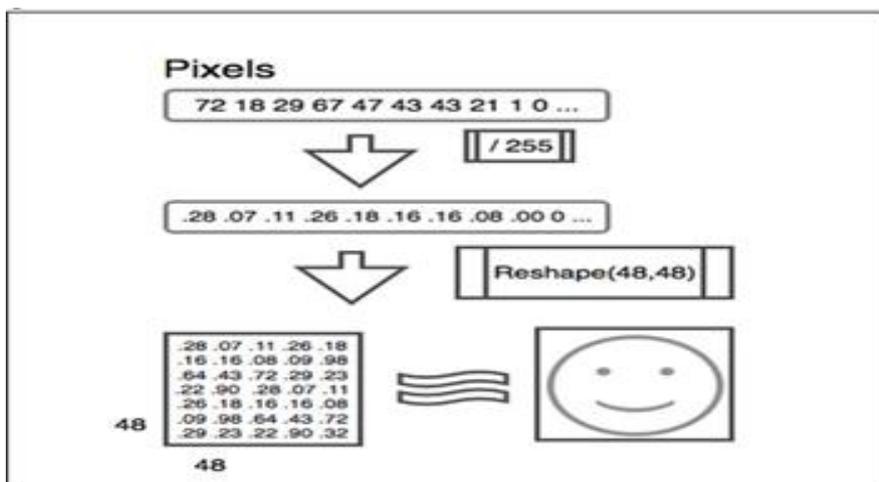


Figure 5. Data Transformation in Facial Emotion Recognition

Each layer will have activation function as leaky. Stochastic gradient descent is used as an optimizer in complying the CNN. The model which is built by this method is saved (the classes are saved in json whereas weights are saved in .h5py format). This saved model is loaded into another program. Haarcascade frontal face classifier is used to detect face from input. Features are extracted from detected face and are given to that loaded model to predict class of image. The current expression of user is detected.

3. Sarcasm Detection

1. Results of the module sentiment analysis from text and facial emotion recognition are integrated for detection of sarcasm.
2. If both the modules have same result, then there is no sarcasm and the sentence will be termed as non-sarcastic.
3. If result of modules are not same, then there is presence of sarcasm and termed it as a sarcastic.

Table 3. Sarcasm detection

Sentiment Analysis From Text	Facial Emotion Recognition		
		Positive	Negative
	Positive	True Statement	Sarcasm Detected
Negative	Sarcasm Detected	True Statement	

5. IMPLEMENTATION RESULT

We evaluated the performance for modules of sentiment analysis and facial emotion recognition individually and integrated module of both modules. Accuracy for sentiment analysis from text is 82.4%, for facial emotion recognition is 92.4% and for sarcasm detection is 80.4%.

Table 4. Accuracy of Unit Module

Unit Module	Accuracy
Sentiment Analysis from text	82.4%
Facial Emotion Recognition	92.24%

Table 5. Confusion Matrix for Sentiment Analysis

	Predicted Positive	Predicted Negative	
Positive	6655	1261	7916
Negative	1541	6503	8044
	8196	7764	15960

Null accuracy is 50.40% and accuracy score of this trained model is 82.44%, model is 32.04% more accurate than null accuracy. From the above classification reports, we can see that model has slightly higher precision in negative class and higher recall in positive class. But this averages out by calculating the F1 score, and for both classes, we get the almost same F1 score for both positive and negative class.

Table 6. Accuracy of Sentiment Analysis

	Precision	Recall	F1-Score	Support
Positive	0.84	0.81	0.82	8044
Negative	0.81	0.84	0.83	7916
Average	0.82	0.82	0.82	15960

Accuracy for facial emotion recognition is 92.4%. Stochastic gradient descent is used as an optimizer in complying the CNN with learning rate = 0.055 and decay of = 1e-5. Performance of emotion recognition is increased using 10 epochs.

Table 7. Accuracy of Facial Emotion Recognition

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Recall	0.8131	0.9224	0.9142	0.9610	0.9174	0.9622	0.9535
Precision	0.9161	0.9553	0.9106	0.9673	0.8884	0.9548	0.8801

6. Conclusion

We concluded that sarcasm can be detected on various aspects like characteristics of sarcasm, type, tuple representation and sarcasm as a dropped negation. We observed a very good classification accuracy for random division of data and satisfactory classification accuracy for natural division of data. The classification accuracy was influenced by user plays original facial expression.

References

1. Aditya Joshi, Pushpak Bhattacharya and Mark J Carman, "Automatic Sarcasm Detection: A Survey", ACM Computing Surveys, pp. 1-17 2017.
2. Rachel Rakov and Andrew Rosenberg, "A System for Sarcasm Detection in Speech", Interspeech, pp. 842-846, 2013.
3. Pranav Goel Aditya Joshi, Pushpak Bhattacharya and Mark J Carman, "Sarcasm Target Identification: Dataset and An Introductory Approach", Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp.2676-2683, 2018.
4. Haseena Rahmath and Tanvir Ahmad, "Sentiment Analysis Techniques- A Comparative Study", IJCEM International Journal of Computational Engineering & Management, Vol. 17, Issue 4, 2014.
5. Apoorv Agarwal, Boyi Xie, Ilia Vovsha and Owen Rambow. "Sentiment Analysis of Twitter Data", Department of Computer Science Columbia University.
6. Pawel Tarnowski, Marcin Koldziej and Andrzej Majkowski, "Emotion Recognition using Facial Expressions", Elsevier International Conference on Computational Science, ICCS 2017, pp.1175-1184, 2017.
7. Bincy Thomas, Vinod P and Danya K A, "Multiclass Emotion Extraction from Sentences", International Journal of Scientific Engineering Research, Volume 5, Issue 2, pp.12-15, 2014.
8. Ankush Mittal and Amarvir Singh, "Sentiment analysis Using N-gram Algo and SVM classifier", Volume 5, Issue 4, 2017.
9. Namrata Godbole, Manjunath Srinivasiah and Steven Skiena, "Large-Scale Sentiment Analysis for News and Blogs", ICMWA Stony Brook University, NY 11974-4400 USA.
10. Payal B. Awachate and Prof. Vivek .P .Kshirgar, "Improved Twitter Sentiment Analysis using N gram Feature selection and combinations", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 9, pp.154-157, 2016.
11. Shiv Naresh Shivhare and Prof.Saritha Kethawat, "Emotion Detection from Text", Data Mining and Knowledge Management Process (DKMP 2012), New Delhi, pp.1-7, 2012.
12. Hema Krishnan, M. Sudheep Elayidom and T. Santhanakrishnan, "Emotion Detection of Tweets using Nave Bayes Classifier", International Journal of Engineering Technology Science and Research, Volume 4, Issue 11, pp.457-562, 2017.
13. Romana Rahman, Tajul Islam and Md. Humayan Ahmed, "Emotion detection from Text and Emoticons", London Journal of Research in Computer Science and Technology, Volume 17, Issu 3, pp.1-8, 2017.
14. N. Azmina M. Zamani, Siti Z. Z. Abidin, Nasiroh Omar and M. Z. Z.abiden, "Sentiment Analysis: Determining People's Emotions in Facebook", 13th International Conference

- on Applied Computer and Applied Computational Science, 2014.
15. Saima Aman and Stan Szpakowicz, "Identifying Expressions of Emotion in Text", Springer Identifying Expressions of Emotion in Text, pp.196-205, 2007.