Sentiment Analysis for the Detection of Sarcastic and Ironic Tweets

Susmita Sadanand, Govardhan Hegde K.

Abstract: ***This paper aims at detecting sarcasm and irony tweets based on the application of natural language processing and sentiment analysis. These days twitter has become most widely used social media. Most of the tweets generated affect people's mental health and thought process. Even though many tweets have a positive effect a few of them are targeted towards people for bullying and hurting them. So it is necessary that we filter the tweets and identify the negative ones so that people may have a positive experience on this platform. In order to do this, this paper provides a methodology that helps in analyzing the sentiments behind the tweets and classify them into positive and negative tweets. Neural Network is used to achieve this. Feature engineering is applied on the dataset and then using Neural Network we try to get the result.***

Keywords: detection of sarcasm and irony, natural language processing, sentiment analysis.

I. INTRODUCTION

Sentiment analysis is the contextual mining of texts which is used to identify and extract subjective information in the source material. It is the process of discovering the emotional tone underlying in a series of words which is used to obtain an understanding of the attitudes, emotions and opinions expressed within an online mention. The use of natural language processing, computational linguistics, text analysis and biometrics for systematic identification, extraction, quantification and study of affective states and subjective information is called sentiment analysis. Sentiment analysis is also known as opinion mining and emotion AI. It is usually conducted to help a business interpret the social sentiment of their brand, service or products while observing online conversations. This analysis is highly useful in monitoring social media as it permits us to obtain an overview of the large group of public opinions behind certain topic. Application of sentiment analysis are powerful and broad. In this study, sentiment analysis of tweets is carried out.

Sentiment analysis which is used to analyze social media is called social sentiment analysis. Social sentiment analysis is an algorithm that obtains the contents of social media the sentiment of social media content, like tweets and analyses the sentiment of it. The algorithm accepts a string as input and returns sentiment rating for "positive", "negative" and "neutral" as output.

Revised Manuscript Received on March 05, 2020.

Susmita Sadanand, M.Tech Student, Department of Computer Science & Engineering, Manipal Institute of Technology, Manipal, Karnataka, India.
Govardhan Hegde K., Asst Professor, Department of Computer Science & Engineering, Manipal Institute of Technology, Manipal, Karnataka, India.

II. LITERATURE REVIEW

Pandey et al. [1] proposed a hybrid clustering method which used Cuckoo Search and K means algorithms. They have used 3 different datasets and the proposed method was compared with the seven existing methods. The proposed method was evaluated using three parameters. They are computational time, accuracy and fitness function value. The proposed method gave the best accuracy compared to the existing methods. A statistical comparison was performed to test the difference between proposed and considered methods for computational time, accuracy and fitness function value using student's t -test with a confidence level of 95%.

FICAMOS et al. [2], after the data pre-processing feature extraction is done by extracting bags of words. The focus of the paper was reduction of the features so as to improve the accuracy of the prediction for topic modelling as well as sentiment analysis. Extracting topics from the samples was the initial requirement of this method and then train the algorithm. And then sentiment estimation is done. 74.09% accuracy was obtained from this method.

Barbieri et al. [3] detected irony as a classification problem using supervised machine learning method to the twitter data. The problem was approached as a binary decision. Hence two tree-based classifiers were used. Random Forest and Decision tree. Implementation was done using Weka toolkit. Seven groups of features were used and each tweet was classified under them. The seven groups are frequency, intensity, structure, synonyms, sentiments, ambiguity and written-spoken. Finally, after the analysis, when the comparison was done against other existing model, the result of the proposed model was better than the existing one.

Peng et al. [4] detected sarcasm in the text. To achieve this baseline model was used. The aim of this paper was to conduct error analysis to pinpoint areas for improvement in the existing model. The Naïve Bayes classifier gave an overall accuracy 62.02%. The Gaussian kernel achieved a testing accuracy of 82.2%. And the final conclusion was that the use of unigrams and bigrams alone are not sufficient in designing an accurate classifier. But when combined with other types like topic modelling, the accuracy is greatly increased.

Hee et al. [5] in the book talked about detecting irony in social media. Baseline classifier was used and the total accuracy obtained was 68.37%. Evaluation was done on the groups in isolation and also in different combined setups. Also, binary classification experiments were used using manually annotated tweets. Combining lexical with syntactic and semantic information sources an accuracy of 70.11% was obtained.

Bindra [6] proposed a system in which the system classified a tweet as ironic or non-ironic through supervised learning approach. The process consisted of text pre-processing, tokenization, removal of stop words, lemmatization, extracting features and running the different types of classifiers against the data set. The accuracy obtained was 0.5089 and 0.4923.

Ren et al. [7], in this paper, sarcastic tweets were detected using neural network models. There are two models used in this paper. First one is neural model using context augmentation by integrating key contextual information (CANN KEY) and second one is neural model using context augmentation by integrating all contextual information (CANN ALL). In CANN-KEY model there are two main parts. A local sub network which is a left part uses information from the target tweet whereas a contextual sub network which is a right part uses information from contextual tweets. The left part contains five layers. The layers are input, convolution, pooling, nonlinear combination hidden and output. The right part is also executed in the same way except that it does not use any convolutional function. In the CANN-ALL model there are six layers. The layers are input, convolution, pooling, nonlinear combination, softmax and output. The result concluded that the proposed neural model using context augmentation gave a better performance when compared with the context based model and state-of-the-art discrete model.

Dimovska et al. [8], this paper discusses about the detection of sarcasm and irony tweets. Initially the data is pre-processed by removing stop words and those terms which in appear less than n files. Feature and model selection is done on the pre-processed data. The classification was done using k-Nearest neighbor classifier, linear support vector machine, logistic regression and non-linear support vector machine using Gaussian kernel. There was no significant change found after the pre-processing of the data by removing stop words and by the replacement of hashtags and URLS. A better performance was obtained when character n-grams and word n-grams was used as features. SVM classifier provided the best results. In their future work, instead of single feature type, combination of feature types are to be considered.

Yavanoglu et al. [9], in this paper, different classification algorithms for sarcasm detection were discussed. Two approaches were considered in this work. Machine learning based and rule-based approach. In machine learning based, supervised learning, semi supervised learning, structured learning, hybrid approach and neural approach were discussed. In rule-based approach, semantic based approach and statistical based approach were talked about in this paper. The result showed that it cannot be generalized as to which algorithm to use as there exists a different nature, shape and also application of sarcasm which exists in real life.

Y. Manohar et al. [10], in this paper, Improvement of sarcasm analysis using corpus-based approach and NLP was done. In their proposed method, main aim was to differentiate tweets into sarcastic and non-sarcastic. In this work the following steps were considered. Data was collected, data pre-processing was performed on the data, classification algorithm was used and system model was used. The model was used to find if the tweet is sarcastic or tweet is non-sarcastic. The model consisted of NLP classifier, tokenization, part-of-speech tagging, corpus-based method and performance measures. The proposed method with the new techniques showed a good result on the real time database of twitter.

T. Jain et al. [11], in this paper, the tweet sentiment is identified first, whether the tweet is positive, negative or both. After that the emoticons are extracted. Based on the phrases and emoticons the tweets are classified into sarcastic or non-sarcastic. The first step in this paper is data extraction and cleaning. The next step is seeding in which a list of phrases with positive sentiment and negative situation are generated which is given as input to the machine learning classifiers for extractions of features and pragmatic extractions. The next steps consist of lexical classifier, machine learning classifier, emoticon classifier, pos-neg recognition and pragmatic classifier. The proposed method increased the performance and the pragmatic classifier further increased the overall accuracy.

Chaudhari et al. [12], this paper is a literature survey about sarcasm detection. In this paper, initially different types of sarcasm were discussed. And then the feature set analysis is done based on the types of sarcasm that occur in the tweet. Features can be represented as a collection of set namely lexical feature, hyperbole feature, pragmatic feature, pattern-based feature, contextual feature, syntactic feature and metaphorical feature. And then the issues in the sarcasm detection was discussed. Some of the issues are issues with data, features extracted and classification techniques. Some of the challenges that were discussed in this paper are difficulty in the detection of sarcasm from text, negative sentiment using positive words, usage of short texts, integration of world knowledge, hyperbole and additional features. Different types of approaches that the paper talked about was rule based approaches, statistical approach, distributional approach, classification and deep learning approach.

Razali et al. [13], this paper talks about the importance of multimodality in the detection of sarcasm for sentiment analysis. In this paper sarcasm detection has been done using three approaches. The first approach is rule-based approaches. The second approach is statistical approach and the third approach is deep learning. In deep learning, algorithms which were used are Convolutional neural network (CNN), deep neural network (DNN) and long short-term memory (LSTM). These approaches were compared with support vector machine and a significant improvement was seen.

Gidhe et al. [14], in this paper, sarcasm detection of statements without the hashtags is done using MLP-BP. In the proposed system, the dataset was pre-processed using tokenization. Then the feature extraction was done on that pre-processed data. Three different categories of features were considered namely structural features, semantic features and affective features. Finally, the statements were classified into sarcastic and non sarcastic using multilayered-perceptron backpropagation algorithm (MLP-BP).

The output of the feature extraction was given to a classifier that is backpropagation feedforward network and classification of sarcastic and non-sarcastic sentence was obtained.

Raghav et al. [15], this paper was a review of automatic sarcasm detection. The dataset was categorized into short text, long text and other dataset. The approaches in which sarcasm was detected was rule based approach, statistical approach, classification approach and profound learning-based approach. Some of the challenges discussed in this paper are challenges with features, working with skewness of dataset. The future work will focus on covering different forms of sarcasm and also detect sarcasm in new languages. The future work will also include exploring feature engineering based on conceptual approach.

Dharwal et al. [16], in this paper, automatic sarcasm detection using feature selection is carried out. The proposed architecture consists of creating training and the test sets, pre-processing of the dataset, feature engineering, feature selection, classification and model evaluation. The different steps used in feature engineering are tokenization, stemming, uncapitalization, sentiments, recognition of pattern. For classification support vector machine, logistic regression, Naive Bayes algorithms were used. The conclusion was that Naïve Bayes method does not produce higher precision and there are high chances of poor results. SVM can provide more accuracy but the only issues are speed and time. Logistic regression can also provide excellent results but not as efficient as support vector machine.

Agrawal et al. [17], this paper is about affective representations for sarcasm detection. The proposed model consists of creating weakly labelled data using distant supervision, affective representation of word and sarcasm detection. In the first part, data is extracted and affect labelling is done. In the second part, i.e., learning affective word representations, recurrent neural networks were used to obtain representations of word. In this phase all the input words are converted to their corresponding vector representation. And finally, sarcasm detection was done. The conclusion was that affective word representation was more acceptable for short text documents like tweets and emotion word representations are better for the detection of sarcasm in long documents.

Sandeepa Kannangara [18], this paper is about fine-grained political opinion polarity classification and sarcasm detection using twitter mining. In this paper, proposed fine grained opinion mining model was called joint entity sentiment topic. The steps were extraction of opinion target, topics discussed and sentiment for the target from each tweet. Then political orientation detection and sarcasm detection in political tweets were done. For the sarcasm detection linguistic and contextual features were used and supervised model was used.

Tsakalidis et al. [19], this paper is about building and evaluating the resources for the analysis of sentiment in the Greek language. Different algorithms were used for different sentiment related tasks namely sentiment analysis, emotion intensity analysis and sarcasm detection. Initially the datasets were collected and then feature extraction was carried out. Extracted feature sets were n-grams, lexicons and word embedding. For the classification the algorithms used were logistic regression, random forest and support vector machine and a better result was obtained.

The accuracy and the efficiency of the sentiment analyzer can be increased by using machine learning algorithms. And also, detection of sarcasm and irony which was not done previously can be done together.

III. METHODOLGY

A. Sarcasm Detection

The design of the proposed methodology is shown below:

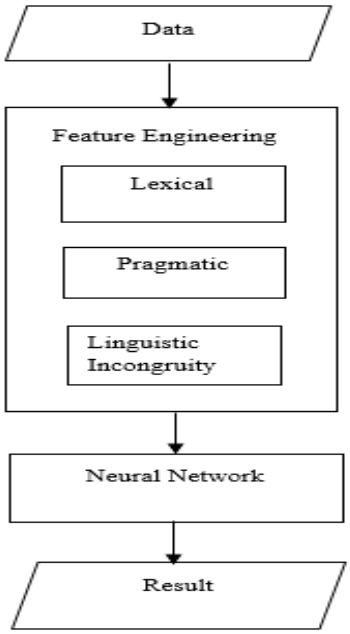


Fig. 1.Design for Irony detection

i. Data

For the sarcasm detection in tweets, the tweets are mostly sampled using the keywords #sarcasm tweets by querying the streaming API and removing all those tweets which are non-English. This is done using the resources available at [20]. But, collection of tweets is very time consuming. Hence, the existing dataset is used, courtesy of [21].

ii. Feature Engineering

In any Natural Language Processing central focus is given for the features. The features selected in turn results in the quality of classification. Features that are chosen and designed carefully helps to improve qualitative and quantitative results.

Sarcasm detection is a difficult task. In most of the cases, sarcasm is embedded into a positive sentiment statement. Another important factor is context which helps in determining if sarcasm is present or not. Thus, in the Natural Language Processing it is a complex task.

The features used can be classified into three categories:

- Lexical
- Pragmatic
- Linguistic Incongruity

1) Lexical Features

N-grams are generally used feature set for the tasks related to NLP in Machine Learning.

There are some words or phrases which a strong indicator for the presence of sarcasm can be. Unigrams is used to extract the lexical features present in the tweets.

A dictionary is created using the training dataset. Each unique word is being mapped onto a particular ID. These ID numbers are the feature numbers. These feature numbers are the frequency of occurrence of a word present in the tweet for which the feature values are being generated. The dictionary will be huge due to the vocabulary available in dataset. From this large vocabulary the tweet would contain only the few words. Thus, for the words that do not occur in the tweet will contain 0's in their respective feature set. The ID's with value 0 can be removed.

2) Pragmatic features

For sarcasm detection grammar also play a major role. Number of emoticons, number of slang laughter expressions, number of capital letters and number of punctuation marks are the features that is used here.

- In social media, emoticons are used to express sentiments. They can be extracted as a feature using UTF-8 encoding. Files that contain emoticons in UTF-8 format can be opened and read using 'codecs' module in Python. These emoticons can be captured using regular expressions.
- Some of the popular slang expressions that are used are 'lol', 'rofl', 'lmao', etc. The number of occurrences of these slang expressions is also a feature. The high occurrence can indicate the presence of sarcasm.
- In most of the cases, people use capitalization to give extra importance to their emotions. For the extra impact sarcasm might also be highlighted. Hence number of capital letters is a feature.
- In a similar way punctuations marks can also be added to create extra impact. And thus, number of punctuation marks is a feature.

3) Explicit Incongruity

The expressions that contain positive sentiment with a negative situation are often sarcasm. For example: 'My finger hurts. Yay!' in this tweet, hurt is a negative word whereas the word yay implies a positive situation and at the end there is an exclamation mark as well which is used to emphasize it. Thus, explicit incongruity is an indication of the presence of contrasting emotions in the tweets which is an implication of sarcasm.

Senti-strength tool is used for this. This tool generates polarity for each word. The generated value lies in the range [-5,5]. Positive value indicates positive polarity and negative value indicates negative polarity. Total number of positive and negative polarity words are counted.

iii. Neural Network

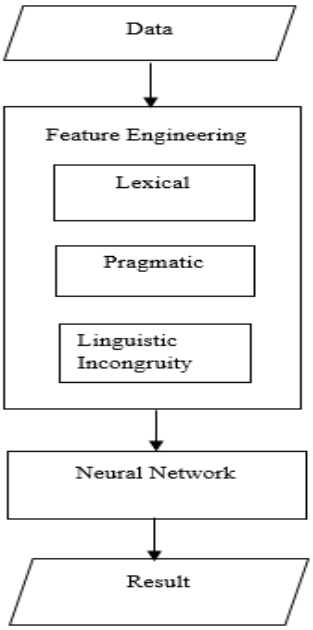
Neural network is a network of simple elements. These simple elements are called artificial neurons. This network takes input, based on the input internal state (activation) is changed and output is produced depending on input and activation.

A neural network is a sorted triple (N,V,w), where N is the set of neurons, V is a set { (i,j) | i,j ∈ N} where elements are called connections between neuron i and neuron j and w is a function. The function is defined as w: V □R which is the weights where w((i,j)) is the weight of connection between neuron i and neuron j.

Here, for the analysis of sarcastic tweets five neurons and one layer is used.

B. Irony Detection

The design of the proposed methodology is shown below:



i. Data

For the sarcasm detection in tweets, the tweets are mostly sampled using the keywords #sarcasm tweets by querying the streaming API and removing all those tweets which are non-English. This is done using the resources available at [20]. But, collection of tweets is very time consuming. Hence, the existing dataset is used, courtesy of [21].

ii. Feature Engineering

1. Lexical Features

Lexical features include 1-, 2-, and 3-grams in both character and word levels. The top 1,000 n-grams based on the term frequency inverse document frequency (tf-idf) values for each n-gram type, have been used. That is, each n-gram appearing in a tweet becomes an entry in the feature vector with the corresponding feature value tf-idf. Also features such as number of words and number of characters are used.

2. Syntactic Features

To tokenize and define part-of-speech tags (POS tags) for all tweets in the dataset NLTK toolkit is used. Then all the POS tags with their respective tf-idf values are utilized as syntactic features and feature values, respectively.

3. Semantic Features

The Lexicons used in tweets are informal and this is the major challenge as it differs from tweet to tweet. This property is not well recognized by syntactic and lexical features. To handle this problem, three approaches are applied to compute tweet vector representations.

Firstly, 300-dimensional pre-trained word embeddings is employed to compute a tweet embedding as the average of the embeddings of words in the tweet.

Secondly, the latent semantic indexing is applied to capture the underlying semantics of the dataset. Here, each tweet is represented as a vector of 100 dimensions.

Thirdly, tweet representation is extracted by applying the Brown clustering algorithm -a hierarchical clustering algorithm which groups the words with similar meaning and syntactical function together. A set of clusters is obtained by applying the Brown clustering algorithm, where each word belongs to only one cluster. The algorithm is run with different number of clustering settings (i.e., 80, 100, 120) to capture multiple syntactic and semantic aspects. For each clustering setting, feature used is the number of tweet words are used in each cluster. After that, for each tweet, the features are concatenated from all the clustering settings to form a cluster-based tweet embedding.

4. Polarity Features

Motivated by the verbal irony by means of polarity contrast, such as "I really love this year's summer; weeks and weeks of awful weather", the number of polarity signals appearing in a tweet is used as the polarity features. The signals include positive words (e.g., love), positive emoji icon, negative words (e.g., awful) and negative emoji icon. Positive and negative words in a tweet are recognized using sentiment dictionary. In addition, the presence of negation word in a tweet(e.g., not, n't) using boolean features.

iii. Neural Network

Neural network used in irony detection is same as the one used in sarcasm detection. The only difference is number of layers used in this two.

IV. RESULT AND DISCUSSIONS

A. Sarcasm Detection

In the Neural network, the parameters used are: Number of neurons=5, Number of hidden layers =1.

The values shown in the table were obtained using the proposed model.

Table- I: Result analysis of Sarcasm Detection

Accuracy	Precision	Recall	F-Measure
0.74	0.79	0.82	0.80

B. Irony Detection

In the Neural network, the parameters used are: Number of neurons=5, Number of hidden layers =2.

The values shown in the table were obtained using the proposed model.

Table- II: Result analysis of Irony Detection

Accuracy	Precision	Recall	F-Measure
0.70	0.61	0.69	0.64

V. CONCLUSION

In this research work, detection of sarcasm tweets and ironic tweets is done using Neural Network algorithm. Initially the dataset is collected. After which feature extraction is carried out. Three different features were considered for sarcasm detection and four different features were considered for irony detection. Finally, the feature vectors were given as the input to neural network algorithm. Algorithm gives the accuracy, precision, recall and F-measure.

REFERENCES

1. Avinash Chandra Pandey, Dharmveer Singh Rajpoot, Mukesh Saraswat, "Twitter sentiment analysis using hybrid cuckoo search method", Information Processing and Management, Vol. 53, pp. 764-779, 2017
2. Pierre FICAMOS, Yan LIU, "A Topic based Approach for Sentiment Analysis on Twitter Data", International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 7, No. 12, pp. 201-205, 2016
3. Francesco Barbieri, Horacio Saggion and Francesco Ronzano "Modelling Irony in Twitter", Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 56–64, Gothenburg, Sweden, April 26-30 2014.
4. Chun-Che Peng, Mohammad Lakis, Jan Wei Pan, "Detecting Sarcasm in Text: An Obvious Solution to a Trivial Problem", Writeup for Stanford CS 229 Machine Learning Final Project, 2015
5. Van Hee, Cynthia, "Can Machines Sense Irony? : Exploring Automatic Irony Detection on Social Media", Ghent, Belgium: Ghent University. Faculty of Arts and Philosophy, 2017
6. Komalpreet Kaur Bindra, Asst Prof Ankita Gupta, "Tweet Sarcasm: Mechanism of Sarcasm Detection in Twitter", International Journal of Computer Science and Information Technologies, Vol. 7 (1), pp. 215-217, 2016
7. Yafeng Ren, Donghong Ji, Han Ren, "Context-augmented convolutional neural networks for twitter sarcasm detection", Neurocomputing 308 (2018) 1–7
8. Jona Dimovska, Marina Angelovska, Dejan Gjorgjevikj and Gjorgji Madjarov, "Sarcasm and Irony Detection in English Tweets", ICT 2018, CCIS 940, pp. 120–131, 2018
9. Uraz Yavanoglu, Taha Yasin Ibisoglu, Setra Genyang Wıcana, "Sarcasm Detection Algorithms", International Journal of Semantic Computing, Vol. 12, No. 3, pp. 457–478, 2018
10. Manoj Y. Manohar, Pallavi Kulkarni, "Improvement Sarcasm Analysis using NLP and Corpus based Approach", International Conference on Intelligent Computing and Control Systems, ICICCS 2017
11. Tanya Jain, Nilesh Agrawal, Garima Goyal and Niyati Aggrawal, "Sarcasm Detection of Tweets: A comparative Study", Proceedings of 2017 Tenth International Conference on Contemporary Computing (IC3), 2017
12. Pranali Chaudhari and Chaitali Chandankhede, "Literature Survey of Sarcasm Detection", IEEE WiSPNET, 2017
13. Md Saifullah Razali, Alfian Abdul Halin , Noris Mohd Norowi and Shyamala C. Doraisamy, " The Importance of Multimodality in Sarcasm Detection for Sentiment Analysis", IEEE 15th Student Conference on Research and Development (SCOReD), 2017
14. Pornima Gidhe, Dr. Leena Ragha, "Sarcasm Detection of non # tagged statements using MLP-BP", IEEE, 2017
15. Shalini Raghav, Ela Kumar, "Review of Automatic Sarcasm Detection", 2nd International Conference on Telecommunication and Networks (TEL-NET 2017), 2017
16. Paras Dharwal, Tanupriya Choudhury, Rajat Mittal and Praveen Kumar, "Automatic Sarcasm Detection using feature selection", IEEE, 2017
17. Ameeta Agrawal, Aijun An, "Affective Representations for Sarcasm Detection", SIGIR '18, July 8–12, 2018
18. Sandeepa Kannangara, "Mining Twitter for Fine-Grained Political Opinion Polarity Classification, Ideology Detection and Sarcasm Detection", WSDM 2018, February 5–9, 2018
19. Adam Tsakalidis, Symeon Papadopoulos, Rania Voskaki, Kyriaki Ioannidou, Christina Boididou, Alexandra I, Cristea, Maria Liakata, Yiannis Kompatsiaris, "Building and evaluating resources for sentiment analysis in the Greek language", Lang Resources & Evaluation, vol. 52, pp. 1021–1044, 2018
20. Stream data using Twitter API, ,https://github.com/guyz/twitter-sentiment-dataset
21. Sentiment dataset for Sarcasm detection, https://github.com/dmitryvinn/twitter-sarcasm-measurement

AUTHORS PROFILE

Susmita Sadanand, MTech Student, Department of Computer Science & Engineering Manipal Institute of Technology, Manipal - Karnataka. She obtained her B.E. degree from NMAM Institute of Technology, Nitte affiliated to VTU Belgaum. Now pursuing MTech in Computer Science and Information Security from Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal. She has worked on IOT as a part of B.E. project. "Contribute to Features of RSA SecurID Access" is her current ongoing MTech project. Areas of interest include Artificial Intelligence, Machine learning, Big data Analytics, Information security.

Govardhan Hegde K., Asst Professor Selection Grade, currently working in Computer Science and Engineering department since 2010. He has 18 years of teaching experience and nearly 1 year industry experience. He is currently pursuing Ph.D. His areas of interest are Image processing, Data structures, parallel programming, operating systems, finite automata etc. He has nearly 20 papers published in various national and international journal and conferences.
https://scholar.google.com/citations?hl=en&user=-DoUpLYAAAAJ&view_op=list_works&sortby=pubdate