IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

RESEARCH PAPER ON SARCASM DETECTION

[1] Srushti Chandak, [2]Ganesh Mehta, [3]Sakshi Kad, [4] Prof. Rishikesh Sutar

[1]Student,[2]Student,[3]Student,[4]Professor
[1]Department of Electronics & Telecommunication Engineering,
[1]SCTR's Pune Institute of Computer Technology University, India

Abstract: One of the concise fields of Natural language processing is Sarcasm Detection which is used to analyse the given text whether the text contains sarcasm or not. It is an important step in text analysis because if sarcasm is present it creates an ambiguity in the actual meaning of the sentence. Hence finding out whether the text is sarcastic is an important step. Even it is not possible for human brains to find whether sarcasm is present or not in some situations which adds more importance to Sarcasm detection. By analyzing sentiment, businesses and organizations can better understand customer opinions, identify areas for improvement, and make data-driven decisions. However, the accuracy of sentiment analysis can be affected by various factors, such as language complexity,sarcasm.The authors of the paper are final-year students from the Electronics and Telecommunication division of Pune Institute of Information Technology, who are undertaking this project as part of their academic requirements.

***Index Terms* - Natural language processing, Sarcasm Detection, Sentiment analysis.**

I. INTRODUCTION

Customer testimonials are crucial to the dialogue process. Customers can share their experiences on e-commerce websites such as Amazon, Flipkart, and Myntra, providing potential buyers with valuable information about product performance. Identifying positive and negative sentiment from product reviews is essential for gaining true insight. This is a computational survey that uses sentiment analysis to extract subjective information from text. One of the biggest challenges of this application is sarcasm detection. Sarcasm is an unconventional way of communicating out of context which may arise an ambiguity.

NLP serves as a bridge between computer language and human language. An important part of NLP that deals with contextual analysis is sentiment analysis.Sentiment analysis is a method for evaluating an author's point of view and figuring out how they see a circumstance. Entity discovery, anaphora detection, parsing, and sarcasm detection are just a few of the significant difficulties that face sentiment analysis. implemented to classify the polarisation of an argumentative document. Text quality could also be determined by sentiment analysis. Sentiment analysis can be used for a variety of analyses. These tests allow you to identify and obtain different emotion states. The use of the phrases and words from the article allows the parser to investigate various single objects. Another name for this method, which involves talking about a specific entity and gathering comments on it, is recommender systems. In sentiment analysis, a variety of classification models and procedures are employed. Gathering of data and performing necessary steps is an important part of the process. Data preparation techniques include tokenization, stemming, lemmatization, and stopword elimination. There are several categorization methods employed, including Support Vector Machines (SVM), Naive Bayes, and Random Forest. This essay will give a succinct summary of the different approaches and tools used to identify sarcastic text in sentiment analysis. These are the steps that make up this paper:
1. Twitter Archiver is used to gather data. There are #sarcasm-containing data found.
2. Preprocessing of data is done. Tweets that have been retweeted and those without any English language are deleted.
3. The polarity of the data, or whether it is positive, negative, or neutral, is determined.
4. Data is trained, evaluated, and accuracy assessed using SVC

The toolset in use is the Natural Language Processing Toolkit. A group of Python-coded tools for statistical natural language processing makes up NLTK. Sample data and graphical displays are included in NLTK. NLTK comes with different inbuilt functions like tokenization, parsing, etc with its library. Data processing is done using Anaconda. Anaconda is the name of the open source distribution of the computer language Python, which is useful for processing large amounts of data. One of the more than 720 open source packages that make up Anaconda is the pre-installed NLTK package. Anaconda comes with Python pre-installed. Sarcasm identification is important in sentiment analysis because without it, the text's sentiment would be wrongly classified. To improve the effectiveness of sarcasm detection, many models have been trained with a variety of characteristics and algorithms. We have various forms of sarcasm detection, including the following:

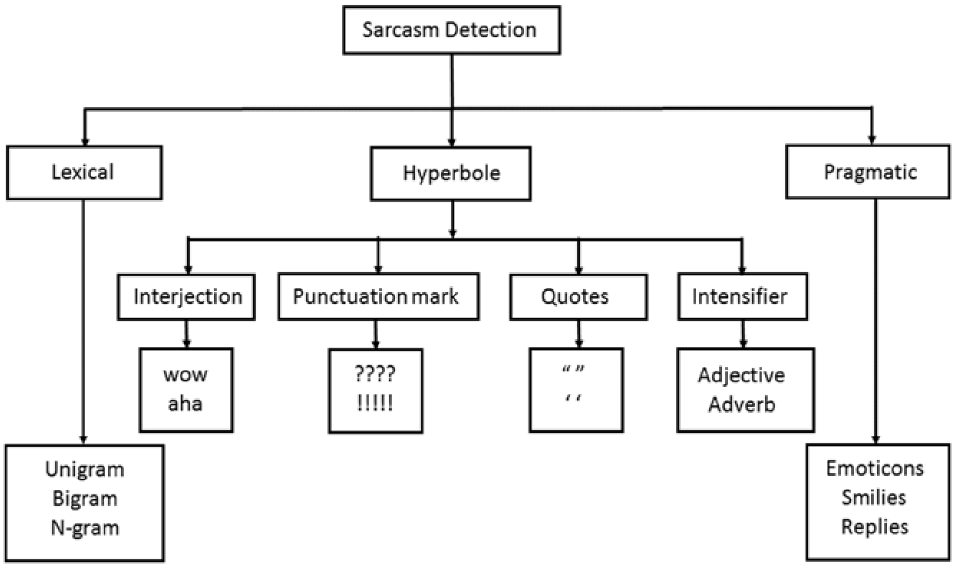

Fig 1. Classification of Sarcasm Detection Method

II. LITERATURE REVIEW

While reading through multiple different papers and websites we found that González-Ibánez et al. [1] in one of his major studies in the field of NLP is his specialized in detecting sarcasm using machine learning techniques. In this survey, the predictive performance of several sets of linguistic features (that is, the presence of unigrams, dictionary-based lexical and pragmatic factors, and the frequency of dictionary-based lexical and pragmatic factors) was compared to the sarcasm obtained from Twitter. It was evaluated using a combination of corpora. We use two classical supervised learners: support vector machines and logistic regression. Dialects are often used to survey emotions. Texts are classified into subjective and objective forms. Emotional dialects are observed using subjective sentences. Polarity identification is done using weighted IDF. A related scenario is POS tags, which may be accompanied by words in reports. Functional taxonomy mostly plays a role in that it designates a taxonomy that expresses sarcasm.

Kunneman et al. [2] analyzed the relatively superior reliability of viewer hashtags on Twitter as gold tags for sarcasm recognition in a study.

Rajadesingan et al. [3] introduced a behavioral modeling scheme for sarcasm detection. On Twitter, Bouazizi and Ohtsuki shared a pattern-based technique for detecting sarcasm. Four main feature sets (sentiment-related attributes, punctuation-related features, syntactic and semantic features, and pattern-based features) were analyzed in this scheme in conjunction with several traditional machine-learning algorithms, which include random forests, support vector machines, the k-nearest neighbor algorithm, and maximum entropy. With an accuracy rate of 83.1%, the empirical analysis revealed that pattern-based feature sets can generate promising results for the activity of sarcasm identification.

X Fang et al. [4] used a dataset of amazon product reviews where their evaluation task was to review level and sentence-level categorisation. The approach used was random forest, Naïve Bayes and SVM where Random Forest performed best in sentence-level categorisation. SVM and Naïve Bayes outperformed random forest in review-level categorization were the result.

A Jeyapriya et al. [5] obtained an aspect extraction accuracy: 80.36% and Sentiment orientation accuracy: 92.37% for sentiment analysis using Naïve Bayes algorithm using a superior term counting-based approach.

Amir et al. [6] used a technique where embedding turned text into higher-dimensional word vectors to create user models that capture homophily. These dimensions make it simple to identify relationships between words and similarities between them using basic functions like sigmoid functions.

In order to extract sentiments, Soujanya Poria et al. [7] created models using a pre-trained CNN layer that autonomously trains using a convolutional neural network and trains its sarcasm features. It was noticed that the tone in the tweets changed, suggesting that the text was sarcastic. In order to incorporate sentiment shifting into the paradigm they suggested, they train their model to extract sentiment-specific features. Convolutional neural networks were used for the classification, and the completely connected layer of the CNNs served as the source of the features. To accomplish the final classification, an SVM was fed the extracted features.

The model's results yielded an F1-score of 76.78An attention-based circular neural network that can extract more text features was proposed by Wang et al. [8] These techniques demonstrated how adding an attention mechanism to CNN can increase the precision of text sentiment analysis.

For experiments on the dataset, MingShan Wang [9] used TF IDF. With the aid of a bag of words, review ratings are predicted. There were used classifiers like the root mean square error and linear regression model. Some related works on sentiment analysis and techniques are listed above. Taking into account the works listed above will help us create a model that is more accurate.

III.EXISTING METHODS

On the basis of our survey, numerous methods for sarcasm detection are discussed. In general, rule-based, statistical, distributional, classification, and deep learning methodologies can be used for sentiment analysis. A thorough explanation of each step is provided below.

1. Rule-Based Approach : In this method to recognise sarcasm, specific evidence is captured in terms of rules. The sarcastic cues are what determine whether these rules be applied. For instance, adding more than one letter to a word is a way to convey irony. The benefit of this method is that it enables the study of faults relating to various rules.

2. Statistical Approach : In this approach, the sentence's content is given more weight than the grammar and sentence structure. For instance, we give more weight to the words that are actually in the sentence. The statistical approach emphasizes the use of various combinations, including pragmatic and lexical ones. This method is used by models like SVM and Naive Bayes, which are also more effective because most test scenarios are covered when utilizing various combinations.

3. Distributional Approach : In this method, the distributional hypothesis is taken into account. It asserts that words often have similar meanings when they are used in the same context. The situational course that is learned from the data substitutes for the meaning of a target term. Geometric methods like cosine similarity applied to context vectors allow for the measurement of the similarity in meaning..

4. Grouping Approach : Finding the intended word or sentence's sarcastic or literal intent typically involves using categorization systems. Models like Random Forest, Decision Tree, and Support Vector Machines are employed. When fresh data is fed into the model, it is placed into the appropriate group after dividing the entire dataset into n groups.

5. Deep Learning Approach : A subset of a larger family of machine learning approaches, deep learning-based sarcasm recognition systems are now becoming available. These depend on learning data representations and utilising high-level knowledge abstractions with the aid of a deep graph. For example Convolutional neural networks.

IV.LIMITATIONS OF SENTIMENT ANALYSIS

There are several limitations to sentiment analysis that need to be considered when using this technique:

1. Contextual ambiguity: When the meaning of the words used in a text is highly dependent on the context in which they are used, sentiment analysis models may not be able to detect the sentiment accurately. For instance, depending on the context, the term "sick" can have either a positive or bad connotation.

2. Domain specificity: When a text contains terminology or references from a different domain than the model, sentiment analysis algorithms might not be able to recognise the sentiment being represented in the text with enough accuracy. For instance, when used on political social media posts, a sentiment analysis model that was trained on product reviews may not perform effectively.

3. Subjectivity: Sentiment analysis is fundamentally subjective since various people may perceive the emotional tone or attitude portrayed in a text differently. Conflicts over the veracity of sentiment analysis findings may result from this.

4. Language complexity : Texts containing complicated sentence patterns or figurative language, such sarcasm, irony, or metaphor, may be difficult for sentiment analysis models to discern.
.
5. Data quality : The quality and quantity of the training data used to build the model has a significant impact on how accurate the results of sentiment analysis are. The sentiment analysis model could not work effectively with new or untested data if the training data is biased or has a narrow scope..

Overall, while sentiment analysis is a powerful tool in NLP, it is important to consider its limitations and use it in conjunction with other NLP techniques and human judgment to ensure accurate results.

V. PRELIMINARIES

The sarcasm detection engine, stop word removal, tokenization, part of speech tagging, parsing, and sentiment analysis are all covered in this section.

5.1 Sarcasm Detection Engine

The detection engine classifies the reviews into three groups: positive, negative, and neutral. Actual positive, actual negative, and sarcastic emotion are further subdivided into positive, negative, and neutral tweets. Real positive tweets, real negative tweets, and tweets that are hard to identify whether they are positive or negative are all included in the category of "actual positive tweets," "actual negative tweets," and "sarcastic tweets," respectively. Figure 2 depicts the sarcasm detection engine.

The answer is positive and sarcastic, which implies the sentiment was positive of the review and expressed in a sarcastic manner, hence the ultimate result displayed will be negative. If the review is indeed positive, however, the result is reported as positive. The same was done with unfavorable reviews. Whatever the tone of the evaluation in the case of neutral, the outcome will always be displayed as neutral.

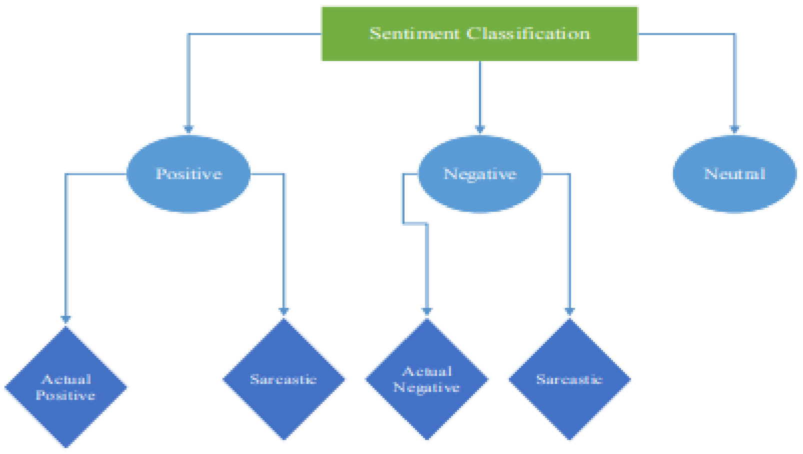

Fig 2. Classification of sentiment of reviews

5.2 Removal of stop words

Python comes with inbuilt library which includes more than 2000 stop words. These dictionary contains stop words which are removed from the sentence if they are present.
"Hey welcome to the new world" for example the words like to, the, this are stop words. After stop words are removed, we are left with "Hey welcome new world.' These word do not add any additional meaning to the sentence but they are important in terms of grammatical formation. For determine the sentiment we do not require these words and hence they are removed from the sentence.

5.3 Tokenization

A text is tokenized when it is divided up into smaller pieces known as tokens. Tokenization is a key activity in natural language processing (NLP), and it is the first step in many NLP applications, such as text categorization, information retrieval, machine translation, and sentiment analysis.

The most common tokenization technique is word tokenization, which involves breaking down a sentence into individual words. For example, the sentence "Hey welcome to the Python Program" would be tokenized into the following words: "Hey", "welcome", "to", "the", "Python", "Program".

Tokenization, however, can also be done at several granularity levels, including character-level, subword-level, and sentence-level. In order to capture morphological information, subword-level tokenization, for instance, divides words into smaller components like prefixes and suffixes rather than breaking them down into individual characters at the character level.

Tokenization can be carried out using a variety of ways, including statistical models and rule-based approaches. While statistical models use machine learning algorithms to learn patterns in the text data and identify tokens, rule-based methods use predefined rules to identify tokens, such as whitespace and punctuation.

5.4 POS Tagging

The natural language processing (NLP) approach known as part-of-speech (POS) tagging includes detecting and classifying the parts of speech of each word in a phrase, including the noun, verb, adjective, adverb, preposition, conjunction, and interjection. A critical step in NLP is POS tagging because it offers vital details about the grammatical structure of a phrase. These details are crucial for a number of downstream tasks, including named entity recognition, sentiment analysis, and machine translation. The right POS tag for each word in a phrase is often predicted using statistical models or rule-based techniques that employ contextual information.

5.5 Parsing

A natural language processing (NLP) technique called parsing involves examining a sentence's grammatical structure in order to determine its meaning. To put it another way, parsing is the process of separating a sentence into its component pieces and figuring out how they relate to one another.

Constituency parsing and dependency parsing are the two primary forms of parsing used in NLP. Constituency parsing is the process of dissecting a sentence into its component phrases, such as noun and verb phrases, and then describing the sentence's hierarchical structure using a structure resembling a tree called a parse tree. On the other hand, dependency parsing identifies the relationships between the words in a sentence, such as the subject-object and modifier-head ties, and uses directed links between words called dependencies to express those interactions.

Different methods, including rule-based approaches, statistical models, and neural networks, can be used for parsing. While statistical models use machine learning algorithms to discover patterns in the text data and determine the most likely parse tree or dependency structure, rule-based methods use predefined grammar rules to parse a sentence. On the other hand, neural networks make use of deep learning techniques to discover the underlying patterns and connections in the text input and produce dependency or parse trees.

Information retrieval, machine translation, and question answering are just a few NLP applications that heavily rely on parsing. Parsing enables a computer to comprehend the meaning of a sentence and extract pertinent information from it by examining the sentence's grammatical structure.

VI. PROPOSED APPROACH

The goal of the suggested approach is to categorize the sarcastic tweets as good, negative, or neutral. Womens Clothing data is collected using different types of datasets. The accuracy and implementation of the Naive Bayes classifier, Stochastic Gradient Descent(SGD) and Support Vector Machine(SVM) classifier will be compared in order to classify the reviews. The Natural Language Toolkit's installed package, WordCloud is used to preprocess the twitter data. Tokenization, part of speech tagging, and parsing are all part of the preprocessing procedures. With the aid of Python programming, the stop words are eliminated. The 2400 stop words in the stopwords corpus, which is provided with NLTK, were used. The detailed implementation of the algorithms is discussed below.

We collected data from different sources to which we applied some data preprocessing. It includes stop words removal using WordCloud function from NLTK library Python. There are 11 different parameters which can be tuned.

1. width: The width of the word cloud image.
2. height: The height of the word cloud image.
3. background_color: The color of the word cloud's background.
4. min_font_size: The minimum font size of the words in the word cloud.
5. max_font_size: The maximum font size of the words in the word cloud.
6. max_words: The maximum number of words to display in the word cloud.
7. stopwords: A set of words to be excluded from the word cloud, such as common stop words like "the" and "and".
8. mask: A numpy array or image that defines the shape of the word cloud. The word cloud will be generated within the shape defined by the mask.
9. colormap: The color map used for coloring the words in the word cloud.
10. relative_scaling: A value that controls the scaling of the word frequencies. A larger value will result in a more equal distribution of word sizes in the word cloud
11. prefer_horizontal: A value that controls the orientation of the words in the word cloud.

We kept background_color='white', stopwords=stopwords, max_words=250, max_font_size=30, scale=2, random_state=5 for training of our model. The next step was to vectorize the sentence into words so that each word can be represented into numerical data.

Algorithm I: A well-liked training approach for machine learning models, including sentiment analysis models, is stochastic gradient descent. SGD classifier contains 9 different parameters.Commonly used parameters of the SGD classifier for sentiment analysis:

1. loss: Commonly used loss functions for sentiment analysis include hinge (linear Support Vector Machine), log (logistic regression), and modified_huber (smoothed hinge loss). Loss function is popularly used for training of model
2. penalty: The type of regularization used for preventing overfitting. Commonly used regularization types include l2 (ridge regression) and l1 (Lasso regression).
3. alpha: The regularization parameter that controls the strength of regularization. Stronger regularization and overfitting are prevented by a higher value of alpha.
4. learning_rate: The learning rate used for updating the model weights. Commonly used learning rate schedules include constant, optimal, and adaptive.
5. n_jobs : The n_jobs parameter can be useful when training large models on large datasets, as it can significantly reduce the time required for training by parallelizing the computations across multiple CPU cores.
6. max_iter: The maximum number of iterations for training the model. Training stops when either the maximum number of iterations is reached or convergence is achieved.

7. tol: Training stops when the change in the objective function is below the tolerance.
8. class_weight: The class_weight parameter can be set to a dictionary of class weights, where each key is a class label and each value is the weight assigned to that class. This can be useful when the frequency of each class is known or when certain classes are considered more important than others.
9. random_state: The random seed used for reproducibility.

For best result we kept class_weight='balanced',n_jobs=-1 while other parameters were kept to default values and we obtain efficiency of 87.3 %.

Algorithm II: During the training of Support Vector Classifier(SVC) model we used linear SVC by keeping its class weight to balanced and remaining to their default values. We obtain an overall accuracy of 90% for SVC Model. To optimize the SVC model, parameter tuning is an important step. Here are some tips for tuning the parameters of an SVM model for sentiment analysis:

1. Regularization parameter (C): The trade-off between increasing the margin and reducing the classification error is controlled by the regularisation parameter. While a large value of C produces a narrower margin and more misclassifications, a small value of C produces a wider margin and fewer misclassifications. With the use of methods like grid search or random search, the value of C can be adjusted.
2. Kernel type: The choice of kernel type depends on the nature of the data and the complexity of the decision boundary. The most commonly used kernels for sentiment analysis are linear, polynomial, and radial basis function (RBF) kernels. The kernel type can be specified as a parameter when creating the SVM model.
3. Kernel parameters: The kernel parameters control the shape of the decision boundary and are specific to each kernel type. For example, the polynomial kernel has a parameter for the degree of the polynomial, while the RBF kernel has a parameter for the width of the Gaussian. The kernel parameters can be tuned using techniques such as grid search or random search.
4. Class weighting: In some cases, the data may be imbalanced with respect to the classes, meaning that one class has more examples than the other. In such cases, it may be beneficial to assign different weights to the classes to ensure that the SVM model learns equally from both classes. The class weighting can be specified as a parameter when creating the SVM model.
5. Feature selection: The choice of features can greatly affect the performance of the SVM model. It is important to select features that are informative and discriminative for the sentiment classification task. Feature selection techniques such as information gain or chi-squared test can be used to identify the most relevant features.
6. Cross-validation: A method for determining the SVM model's performance using hypothetical data is cross-validation. The data is divided into training and validation sets, and the model's performance is assessed using the validation set. To produce a more reliable assessment of the performance, the procedure is iterated several times using various data splits. For the best-performing model, the hyperparameters can be tweaked via cross-validation.

Algorithm III: During the Naive Bayes Model data preprocessing was carried by Tokenizing the data followed by data cleaning which included stop words removal followed by POS Tagging and lemmatization. Different lemmatizers are available like WordNet, Stanford, Spacy, TextBlob, Pattern. Each comes with different types of capabilities. We used WordNet lemmatizer for best results.There are several parameters that can be used to configure a Naive Bayes classifier in scikit-learn:

1. alpha: This is a smoothing parameter that is used to avoid zero probabilities when calculating the likelihood of a feature given a class. A higher value of alpha results in more smoothing, which can help to reduce overfitting. The default value is 1.0.
2. fit_prior: This is a boolean parameter that indicates whether or not to learn class priors from the training data. If set to True, the classifier will estimate the prior probability of each class based on the training data. If set to False, the classifier will use a uniform prior for all classes. The default value is True.
3. class_prior: This is an optional parameter that allows you to specify prior probabilities for each class. If this parameter is not specified, the classifier will learn class priors from the training data. If this parameter is specified, the classifier will use the provided prior probabilities for each class.
4. binarize: This is a threshold parameter that is used to binarize (convert to 0 or 1) the input features. If the value of a feature is less than the binarize threshold, it is set to 0; otherwise, it is set to 1. This parameter can be useful when working with continuous data, but it can also result in information loss. The default value is None, which means that no binarization is performed.

The choice of parameter values can have a significant impact on the performance of the Naive Bayes classifier. We kept default values for all parameters. We obtained 85% accuracy using this method.Overall the SVC Classifier gave the best accuracy which was close to 90%.

VII. RESULTS

From the various algorithms that we implemented, namely, Support Vector Machine(SVM), Naive Bayes, Stochastic Gradient Descent(SGD), SVM provided us with the best accuracy as shown in the figure below.

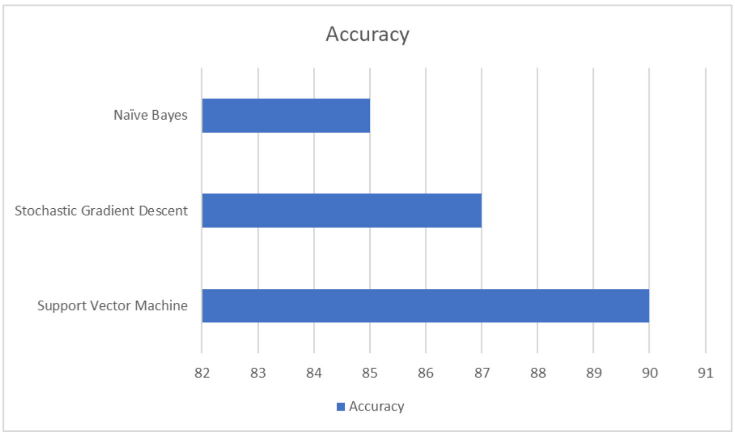

Fig 3. Accuracy of algorithms

We concluded that tuning the parameters of an SVM model for sentiment analysis involves selecting the appropriate kernel type, kernel parameters, regularization parameter, class weighting, and feature selection, and using techniques such as grid search, random search, and cross-validation to optimize the hyperparameters.

VIII. ACKNOWLEDGMENT

We would like to take this opportunity to express our heartfelt gratitude to all those who have contributed in completion of this project. First and foremost, we would like to thank Prof. Rishikesh Sutar, our esteemed mentor, for his invaluable guidance and support throughout the project. His expertise, insights, and constant encouragement were instrumental in shaping our research and helping us overcome the challenges we encountered. We would also like to extend our sincere thanks to our principal, Dr. S. T. Gandhe, and the entire E&TC department staff for their unwavering support and encouragement. Their assistance, both technical and moral, was invaluable in enabling us to carry out this project to the best of our abilities. We are grateful to all the participants who generously gave their time and shared their insights and experiences with us. Without their active cooperation, this project would not have been possible. We would also like to thank our friends and family for their constant encouragement and understanding throughout this endeavor. Finally, we would like to express our appreciation to the entire academic community for fostering an environment of learning and growth. This project has given us the opportunity to expand our knowledge, improve our skills, and gain valuable insights into the field of research. We hope that our work will contribute in some small way to the advancement of knowledge and understanding in our chosen field.

REFERENCES

[1] Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder,"Identifying sarcasm in Twitter: a closer look," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:Human Language Technologies: short papers-Volume 2, Association for Computational Linguistics, 2011

[2] Florian Kunneman, et al., "Signaling sarcasm: From hyperbole to hashtag," Information Processing Management, vol. 51, no. 4,pp. 500–509, 2015.

[3] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu, "Sarcasm detection on twitter: A behavioral modeling approach," in Proceedings of theEighth ACM International Conference on Web Search and Data Mining,ACM, 2015

[4] Xing Fang, Justin Khan,"Sentiment analysis using product review data," Journal of Big Data 2,Article no. 5, 2015

[5] A. Jeyapriya, C. S. Kanimozhi Selvi, "Extracting aspects and mining opinions in product reviews using supervised learning algorithm," 2nd International Conference on Electronics and Communication Systems,2015

[6] Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mario J´ Silva, "Modeling context with user embeddings for sarcasm detection in social media", arXiv preprint arXiv:1607.00976, 2016.

[7] Poria, S., Cambria, E., Hazarika, D., Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:1610.08815.

[8] Wang YHuang MZhao Let al. (2016) Attention-based LSTM for aspect level sentiment classification[C] EMNLP 2016AustinTexasUSA606- 615.

[9] Text mining for yelp dataset challenge; Mingshan Wang; University of California San Diego, (2017)