International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhanced the Accuracy of Text of a Tweet by Detecting Sarcasm using Transfer Learning

Hiteshkumar Barot[1*], Viral A. Parekh[2], Mehul P. Barot[3]

Submitted: 22/10/2023 **Revised:** 12/12/2023 **Accepted:** 22/12/2023

Abstract: A sentence's polarity can be reversed by the use of sarcasm. Despite the widespread usage of sentiment analysis on social media, identifying and analyzing sentiments is still challenging, given the possibility of sarcasm. Even humans find identifying sarcasm in any given text or sentence difficult. Because of this, sarcasm detection with the help of computers becomes an even more significant and challenging task. Several types of research are done regarding sarcasm detection to detect sarcasm in the given text. Unique words are primarily used in sarcasm, and finding the usage of how these individual words are used in tweets or sentences can help detect sarcasm detection. Our research focuses mainly on detecting sarcasm in the texts and creating state-of-the-art model, by combining two models Naïve Bayes, and Decision Tree(entropy-based approach). In our research, we try to combine the methods of the above two approaches to improve the accuracy of sarcasm detection and create a state-of-the-art method for sarcasm detection in texts primarily tweets of the twitter.

Keywords: *Sarcasm, Sentiment Analysis, Twitter, Transfer Learning, Naive Bayes, Entropy based Decision Tree*

1. Introduction

Because of the openness social media platforms provide to express their feelings, social media platforms are gaining more and more popularity, and their usage is increasing day by day in this era of globalization. It is becoming a tool that people are using to express their feelings openly. Out of many social media platforms like Threads, Instagram, etc., Twitter is a social media platform that is regarded as simple to use and very effective in terms of showcasing feelings and opinions for people. Because of its simplicity and effectiveness, people use Twitter to express their feelings; Twitter has become a hotspot not only for businesses but also for researchers to research the enormous amount of data present in Twitter to perform sentiment analysis and detect sarcasm in the tweets that people post.

The purpose of sentiment analysis is to identify whether a text is positive or negative by conducting a computer study on opinions, attitudes, and emotions expressed through text. In his paper [2], researchers surveyed tweets during the US presidential election. To a surprise, 11% of the Twitter users who participated in the discussion related to this topic were found to use sarcasm in either direct or indirect ways in their tweets. This result demonstrates the usage of sarcasm on Twitter only. Paper[3] performed quantitative analysis on the topics where sarcasm is used and found that sarcasm is little in topics like life, health, and food. In contradiction to it, sarcasm is found frequently in topics like politics and government. According to researchers in [5], sarcasm transmits emotions to have fun for humans.

Sometimes, to be able to detect sarcasm is difficult for humans as well. Making a computer learn to see sarcasm in texts becomes a great task, which comes with great difficulty in making computers detect sarcasm in texts. Machine Learning based approaches have been increasingly used by researchers in the domain of sentiment analysis to detect sarcasm in the texts. NLP(Natural Language Processing) is a new area of study that is used to study and examine the data, similar to how humans examine and understand the data.

In our research, we look deeper into sarcasm detection in texts and improve accuracy by keeping the model architecture simple when combining two models without losing accuracy. In our research, we incorporate the weights of the previously trained model as a reference for training the second model.

1 Research Scholar, Department of Technology and Engineering-CE, C U Shah University, Wadhawan, Gujarat, India
2 Dean-Faculty of Engineering, Monark University, Gujarat, India
3 Department of Computer Engineering, LDRP-Institute of Technology and Research, Gujarat, India
Corresponding Author Email: hitesh.ldrp@gmail.com, https://orcid.org/0009-0001-9873-4402

This research proposes a method to recognize sarcastic tweets with high accuracy. In our approach, we consider two approaches and combine the Naive Bayes method and the entropy-based Decision Tree method to determine whether the tweet is sarcastic or not. To improve sarcasm identification in sentiment analysis and achieve more accuracy, we propose an efficient machine-learning model and showcase our results along with the results of two methods for comparison in the later section of the paper. In this work, we try to incorporate the knowledge of the previously trained model as a reference for training the second model by feeding the weights of the first trained model as a starting point to the second model for training to get rich and diversified feature sets for detecting sarcasm in the tweets. We have trained and tested our Approach on the Twitter datasets published by Twitter, which is considered a benchmark dataset for detecting sarcasm in texts. The main contributions of this paper are as follows:

(1) To detect sarcasm in tweets, we try to enrich the feature extraction process and training of the machine learning model by trying to combine features of two different machine learning models.
(2) At the end of this paper, we show the results of our model in comparison with the results of the Naive Bayes approach and entropy-based Decision–tree approach.

The rest of the paper is structured as follows: Section 2 illustrates the literature survey done on different methods proposed for sarcasm detection. Section 3 explains our proposed methodology. Section 4 tells about our experiments and results and the dataset we have used, along with a comparison with other methods. Finally, we conclude our research in Section 5 with a conclusion.

2. Related Work

Many researchers and organizations have become interested in sentimental analysis and, increasingly, in sarcasm detection. More and more researchers have discovered that even though more efficient models are being developed, many factors need to be taken care of because these factors can make analyzing the result of sarcasm detection less accurate.

According to researchers in the paper [1], one factor that is challenging in sarcasm detection is the delivery of opinion using sarcasm. They did research on this factor and recorded that almost 11% of users using Twitter use sarcasm while giving views. Researchers of [2] categorized sarcasm usage in Twitter tweets into three categories: (i) Sarcasm when a person asks a particular question and does not want to give the correct answer. (ii) Sarcasm is used to entertain others. (iii) Sarcasm to express feelings to others. A sarcasm detection technique was proposed by Shubhodip Saha, Prabhat Ranjan, and Jainath Yadav in [3] and employed the Text Blob package from NLTK for preprocessing. The stages involved in preprocessing include tokenization, removing stop words, and parsing. Rapid mining is used to determine tweet subjectivity and polarity. The accuracy of tweets is determined using the Weka tool using Nave Bayes and SVM classifiers. SVM and Neural Networks are used for sarcasm identification in online review texts [4], employing lexical, pragmatic, linguistic, and situational incongruity features. The paper discusses the performance and comparison of SVM and neural networks. Additionally, cross-domain sarcasm detection is taken into account. The findings of the investigation show that the technique is reliable.

By including the history of tweets and author details, the authors David B. and Noah A. S. [5] improved the classification approach and aided in the classification procedure. The article offers accuracy for several scenarios ranging from 70% and higher. Authors of [6] used the gradient Boost method. It showed the best results out of all the classification algorithms. Researchers [7] used substantial linguistic study and lexical elements, including adjectives, adverbs, and interjections, and showed that they significantly impact the ability to recognize sarcasm [7].

This paper not only drives motivation from the previous works of the researchers, but we in this paper also showcase our proposed methodology along with results and comparison with other research as well.

3. Proposed Methodology

In our research, our main aim is to identify the sarcasm in texts and output whether the text has use of sarcastic language or not with high accuracy. In our work, we try to use the knowledge of previously trained models as a reference and perform tuning of the previously trained model into the new architecture to get better accuracy, rather than training the model entirely from the scratch in less time and with fewer resources.

Our model architecture consists of a total of two models that we have used it to the accuracy of identifying and detecting sarcasm in texts like tweets on Twitter. The first model we have used in our model architecture is a Naive Bayes model [8], and the second model, the final model in our model architecture, is an entropy-based decision tree model[9].

For the first model's training purpose, the Naive Bayes method is used to calculate the polarity of the tweet, whether the tweet is a sarcastic tweet or not. The last activation layer of the model has three neurons in it, where each of the three neurons indicates

One class among the polarity of the class labels respectively. The Class labels of the last activation layer are as follows:

(1) **Positive:** Meaning that the tweet is sarcastic in nature.
(2) **Negative:** Meaning that the tweet is non-sarcastic in nature.
(3) **Neutral:** Meaning that the tweet is neutral in nature.

The weights of each neuron and the weights of the neurons for the forward layers are initialized randomly at the start of the training.

We use the Stochastic Gradient Descent training for back propagating the models errors during the training phase and the weights are modified accordingly. A total of 80 iterations are performed for training, where the learning rate is reduced after the first 30 iterations.

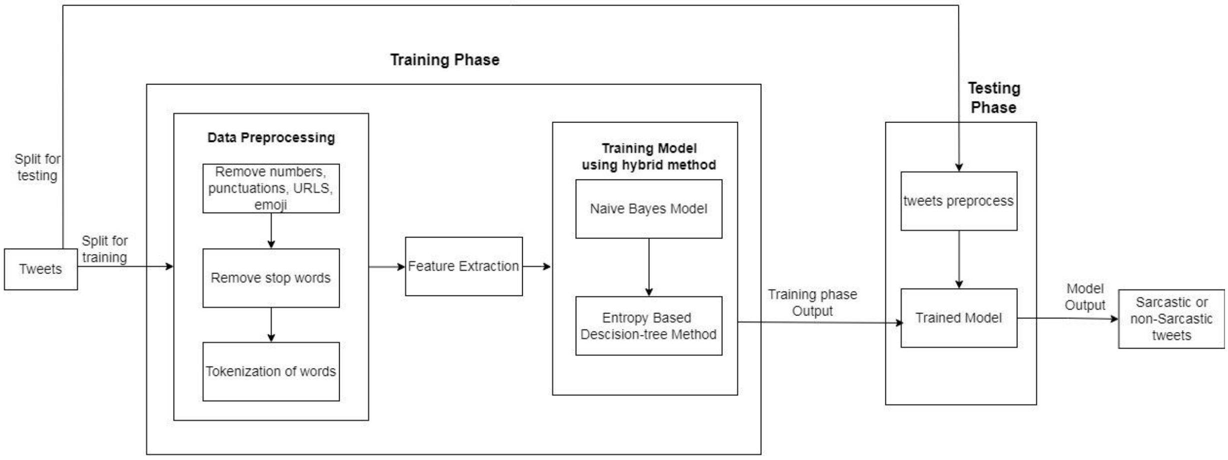

Figure 1. Architecture of proposed method

However, the weights of the second model are initialized with the weights of the first model, which has been trained instead of initializing randomly like the first model. The weights that have been initialized are retrained again and are fine-tuned based on the entropy-based decision tree method[9]. The second model is entropy-based; the generated feature maps are more likely to be robust against changes in the words in the sentence. The architecture of our proposed model is shown in Figure 1: Proposed method's architecture

Data preprocessing and Feature Extraction are two main components of our proposed method's architecture.

Data preprocessing: Its task is to process the raw data, which comes in the form of tweets, and make it suitable for feeding it into the machine learning model in the format that machine learning accepts.

Feature Extraction: Its main task is to take the raw data as an input and process it to transform this raw data into numerical features without making any changes to the original data. It is used to get better accuracy for the input data because of uniformity in the data, which is not visible in raw data.

3.1 Data Preprocessing

Data preprocessing is done on the tweets, which will be fed to our model for higher accuracy.

3.1.1 Text Cleaning

(1) Removing URLs: (Uniform Resource Locators) in the tweet do not add any information in the tweets. They prompt the user to visit the URL for further information. Hence, URLs are removed in data preprocessing.

(2) Removing emoji's: Similar to URLs, emoji's do not add any information or value to the

tweet in terms of sarcasm. So, we remove emoji's from the tweets.

(3) Removing HTML tags: HTML tags such as <html>, <head>, <body>, etc. are removed from the tweets, as they are added to make the tweet more pretty, but actually, these HTML tags do not add any value or information in the tweet. So, we remove html tags from the tweets as well.

(4) Removing punctuation: Punctuations such as #, $, %, &, !, ? etc., are removed from the tweets as they do not add any information in tweets.

(5) Removing quotes: We are eliminating single and double quotes from the tweets as they do not have any information or value in the tweets.

3.1.2 Text Processing

(1) Tokenizing: It is the process of asking an entire sentence as input and splitting it into a list of words called tokens. The token can be words, whole sentences, or even clauses.

(2) Converting to lowercase: We convert all the words of the sentence in lowercase letters to maintain uniformity for all the words and sentences fed into the model for learning and testing purposes because a model can be case-sensitive and produce different results for lower and upper-case words.

(3) Removing stop words: Stop words such as a, an, the, in, to, so, on, etc. are removed from the tweets because they significantly do not have any value or information in the tweet. So, we removed the stop words from the tweets.

(4) Part-of-speech tagging: This method matches a word to its related class, which is then used in the learning process of the model. Part-of-speech taggers take a series of words as input and produce a list of tuples as output, where every word is connected with the relevant tag.

(5) Lemmatization: In this step, we remove the unnecessary tenses or extra words from the actual word and convert the word into its root form, which is the actual word.

Table 1: shows some examples of the twitter dataset. Where the left column shows the original raw tweet and the right column shows the text after the processing is done.

Table 1. Twitter dataset sample.

Original tweet	**Processed tweet**
Flight has not arrived yet	Flight not arrived
Thanks for the response. We are hopeful!	Thanks response hope
Corona is on world tour, it will come to your door very soon	corona world tour door soon

3.2 Feature Extracting

Extraction of the interjection features and unigram features is done in this feature extraction step.

(1) Unigram feature extraction: Unigram feature extraction extracts single words from the text and returns a collection of single words from the given sentence. It is done when text is to be classified as a sarcastic text or not.

(2) Interjection feature extraction: Interjection feature extraction not only extracts the words from the given sentence but also extracts the intentions or the feelings of the person along with the words. According to the research conducted by the researchers in the [10], the use of word interjection is one of the main characteristics for detecting sarcasm in texts.

4. Dataset, Results And Analysis

In this section, we show the results of our proposed methodology and compare our development with the results of the base models - the Naive Bayes model and the Entropy-based Decision tree model, which we took as reference models.

4.1 Dataset

We have trained our model on the Twitter dataset, considered a benchmark dataset in the sarcasm detection domain. The Twitter dataset consists of 1.6 million rows in the dataset, where each row in the dataset is a unique tweet. The Twitter dataset consists of various tweets in different domains ranging from food, health, lifestyle, politics, and many others. The Twitter dataset is a pre-labeled dataset where every tweet is labeled as sarcastic, neutral, or non-sarcastic. The Twitter dataset is openly available from the kaggle website.

We evaluate our approach on the Twitter dataset for testing purposes. Along with calculating the accuracy for the correctly classified labels, we also calculate the misclassified labels for the tweets as a comparison parameter to show the robustness of our proposed methodology with other models.

4.2 Results

4.2.1 Naive Bayes Model

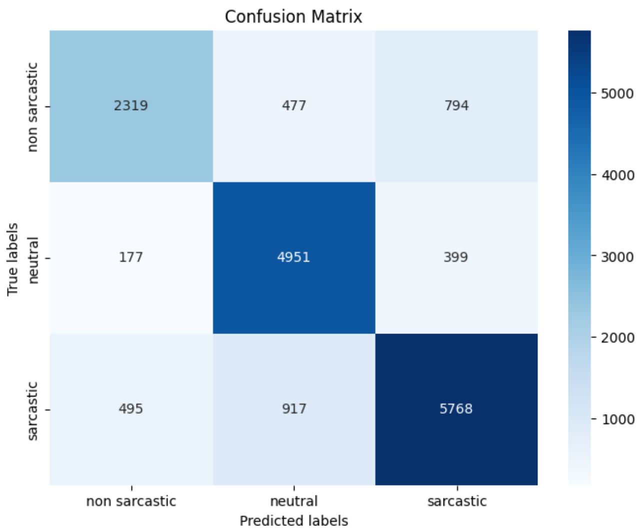

Figure 2. Confusion-Matrix for Naive Bayes

Figure 2, shows the confusion matrix for the Naive Bayes model. The confusion matrix shows the number of tweets which has been classified. The labels on which the tweets have been classified are as follows:

(1) **Negative:** A negative title for a tweet means that the tweet is non-sarcastic.
(2) **Neutral:** Neutral label for a tweet means that the tweet is neutral is nature.
(3) **Positive:** A positive label for a tweet means that the tweet is sarcastic.

From Figure 2 we can see that the Naive Bayes model is correctly able to detect the tweets, and the results are as follows:

(1) Correct Non-Sarcastic: 2319
(2) Correct Neutral: 4951
(3) Correct sarcastic: 5768

Naive Bayes, along with correctly identifying the tweets as sarcastic, non-sarcastic, and natural tweets, also has misclassified tweets and the results are as follows:

(1) False non-sarcastic but were neutral: 477
(2) False non-sarcastic but were sarcastic: 794
(3) False neutral but were non-sarcastic: 177
(4) False neutral but were sarcastic: 399
(5) False sarcastic but were non-sarcastic: 495
(6) False sarcastic but were actually neutral: 917

4.2.2 Entropy-based Decision Tree Model

The figure 3 shows the confusion matrix for the entropy based decision model. The confusion matrix shows the number of tweets which has been classified. The labels on which the tweets have been classified are the same, as described in the naive bayes section.

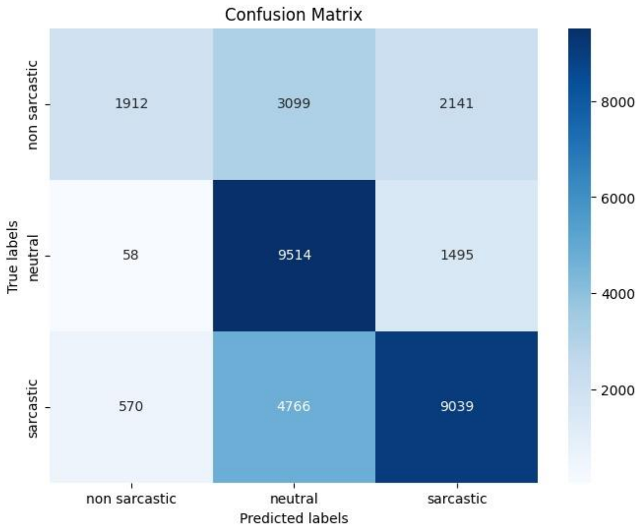

Figure 3. Confusion-Matrix for entropy-based decision Tree

From Figure 3, we can see that the entropy-based decision tree model is correctly able to detect the tweets, and the results are as follows:

(1) Correct non-sarcastic tweets: 1912
(2) Correct neutral tweets: 9514
(3) Correct sarcastic tweets: 9039

Entropy-based decision tree, along with correctly identifying the tweets as sarcastic, non-sarcastic, and neutral tweets, also has misclassified tweets, and the results are as follows:

(1) False non-sarcastic but were neutral: 3099
(2) False non-Sarcastic but were sarcastic: 2141
(3) False neutral but were non-sarcastic: 58
(4) False neutral but were sarcastic: 1495
(5) False sarcastic but were non-sarcastic: 570
(6) False sarcastic but neutral: 4766

From the above Figure 2 and Figure 3, we can see that Naive Bayes out performs the entropy-based decision tree model because the Naive Bayes model is a probabilistic model that is more robust in generating feature maps compared to the decision tree model, which is formed by a grouping of weight assessments which are performed individually on every feature.

4.2.3 Our Proposed Methodology

The Figure 4 shows the confusion matrix for our proposed methodology model. The confusion matrix shows the number of tweets which has been classified. The labels on which the tweets have been classified are the same, which has been described in the naive bayes section.

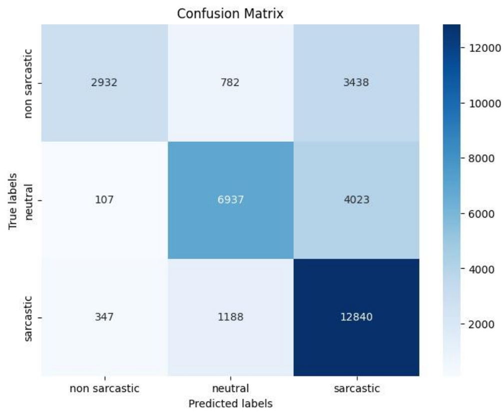

Figure 4. Confusion-Matrix our proposed methodology

From the Figure 4 we can see that our proposed methodology model is correctly able to detect the tweets, and the results are as follows:

(1) Correct non-sarcastic: 2932
(2) Correct neutral: 6937
(3) Correct sarcastic: 12840

Our proposed method, along with correctly identifying the tweets as sarcastic, non-sarcastic, and neutral tweets, also has misclassified tweets and the results are as follows:

(1) False non-sarcastic but were neutral: 782
(2) False non-sarcastic but were sarcastic: 3438
(3) False Neutral but were non-sarcastic: 107
(4) False neutral but were sarcastic: 4023
(5) False sarcastic but were non-sarcastic: 347
(6) False sarcastic but were actually neutral: 1188

4.2.4 Results Comparison

Figure 5 shows the chart for showing the accuracy comparison of our proposed methodology with Naive Bayes and entropy-based decision tree models, which have been considered as our base reference models.

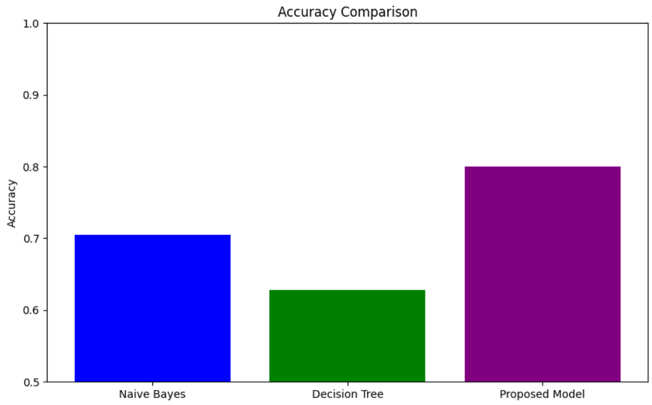

Figure 5. Accuracy chart of all models

From Figure 5, we can see that the accuracy of the Naive Bayes model is around 70%, the accuracy of the entropy-based decision tree is around 63%, and The accuracy of our proposed methodology is about 79%, which is greater than these two methods, Naive Bayes methodology and entropy-based decision tree, which are the base models we have taken as reference.

Table 2. Accuracy Table (Performance)

Model/Method	Accuracy
Naive Bayes	70%
Entropy-based Decision Tree	63%
Proposed Methodology	79%

Table 2 shows the accuracy of each model in a row-wise manner. From this table, we can see that the accuracy of our proposed model is more than that of the Naive Bayes and Entropy-based decision tree models.

Table 3. Misclassified tweets of our model and other models

Model/Method	Actual label	Classified label	Nr. of tweets
Naive Bayes	Non-sarcastic	neutral	477
Naive Bayes	Non-sarcastic	sarcastic	**794**
Naive Bayes	Neutral	non-sarcastic	177
Naive Bayes	Neutral	sarcastic	**399**

Naive Bayes	Sarcastic	non-sarcastic	495
Naive Bayes	Sarcastic	neutral	917
Decision tree	Non-sarcastic	neutral	3099
Decision tree	Non-sarcastic	sarcastic	2141
Decision tree	Neutral	non-sarcastic	**58**
Decision tree	Neutral	sarcastic	1495
Decision tree	Sarcastic	non-sarcastic	570
Decision tree	Sarcastic	neutral	4766
Proposed model	Non-sarcastic	neutral	**782**
Proposed model	Non-sarcastic	sarcastic	3438
Proposed model	Neutral	non-sarcastic	107
Proposed model	Neutral	sarcastic	4023
Proposed model	Sarcastic	non-sarcastic	**347**
Proposed model	Sarcastic	neutral	**1188**

Table 3 shows the misclassified tweets of all the models along with their correct respective labels.

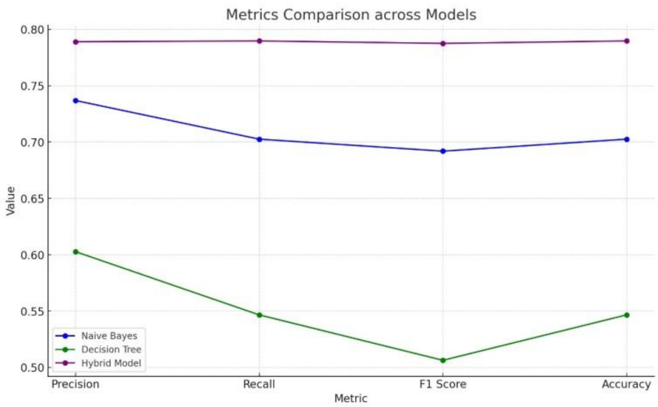

Figure 6. Comparison of metrics for all the measures

Figure 6 shows the comparison of metrics like Precision, Recall rate, F1 score and Accuracy for all three models.

From the Figure 6, we can see that our proposed model out performs than other two models in all the parameters by a very huge margin. It’s showing that our proposed model performs better than other two models in every measuring metrics.

4.3 Analysis

Our proposed methodology outperforms the Naive Bayes and entropy-based decision tree methods because we have combined the good features of both models and incorporated these features into our model for higher accuracy.

From the observation of Table 2, Table 3, Figure 5, and Figure 6, we can see that our model not only provides higher accuracy in detecting tweet labels correctly as they are, but it also has lower misclassifications in predicting tweet labels compared to the other two methods.

We first trained the Naive Bayes model and then used its pre-generated weights to train our second entropy-based decision tree model. After training the Naive Bayes model, our model became robust in generating feature maps, which increased the accuracy for predicting the tweets correctly as they are, as it is a probabilistic-based method. After training our model with an entropy-based decision tree model, our model became robust against the wrong classification of tweets labels because the decision tree used a collective approach to identify weight by considering each feature.

5. Conclusion

In this work, we have proposed a new method for the sarcasm detection problem by taking the transfer learning approach of two different ways into consideration for sarcasm detection. Instead of relying entirely on the weights of Naive Bayes or Entropy-based decision tree, we have tried to combine the knowledge of Naive Bayes and entropy-based decision tree for feature maps and sarcasm detection. The aggregation performed is done on an enriched feature map for sarcasm detection, which made our method more robust towards unique words. Results and analysis show that the transfer learning method is effective, outperforms previous methods, and performs best without needing post-processing methods/techniques.

There is still a large room for improvement in sarcasm detection, not only for texts but also for speeches. We can use relations between the sentence's unique words and enhance the feature maps for better accuracy. We believe these open questions will inspire more future work.

Acknowledgment

This work is not supported by any funding organization nor any institute. I get only direction and technical guidance from the supervisor and co-supervisor who are affiliated with the C U Shah University, Monark University, and LDRP - Institute of Technology and Research.

References

[1] Heredia, B., Prusa, J., & Khoshgoftaar, T. (2017). Exploring the effectiveness of Twitter at polling the United States 2016 presidential election. In Proceedings of the IEEE 3rd International Conference on Collaboration and Internet Computing (pp. 283-290).

[2] Gupta, R., Kumar, J., & Agrawal, H. (2020). A statistical approach for sarcasm detection using Twitter data. In Proceedings of the 4th International Conference on Intelligent Computing and Control Systems (pp. 633-638).

[3] Saha, S., Yadav, J., & Ranjan, P. (2017, July). Proposed approach for sarcasm detection in Twitter. Indian Journal of Science and Technology.

[4] Sharma, S., & Chakravarty, S. (2018). Sarcasm detection in online review text. ICTACT Journal on Soft Computing, 8(3).

[5] Bamman, D., & Smith, N. (2015). Contextualized sarcasm detection on Twitter. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 9).

[6] Prasad, A. G., Sanjana, S., Bhat, S. M., & Harish, B. (2017). Sentiment analysis for sarcasm detection on streaming short text data. In Proceedings of the 2nd International Conference on Knowledge Engineering and Applications (ICKEA) (pp. 1-5). IEEE.

[7] Kreuz, R., & Caucci, G. (2007). Lexical influences on the perception of sarcasm. In Proceedings of the Workshop on Computational Approaches to Figurative Language (pp. 1-4).

[8] Mukherjee, S., & Bala, P. K. (2017, February). Sarcasm detection in microblogs using naive Bayes and fuzzy clustering. Technology in Science. Elsevier.

[9] Sentamilselvan, K., et al. (2021). Sarcasm detection in Twitter. IOP Conference Series: Materials Science and Engineering, 1055, 012105.

[10] Ahuja, R., & Sharma, S. (2020). A review paper on sarcasm detection. In Proceedings of the 3rd International Conference on Advances and Applications in Artificial Intelligence (pp. 1-5).

[11] Baroiu, A.-C., & Trăușan-Matu, Ș. (2022). Automatic sarcasm detection: Systematic literature review. Information, 13(8), 399. https://doi.org/10.3390/info13080399

[12] Kitchenham, B., & Charters, S. (2007, January). Guidelines for performing systematic literature reviews in software engineering (Vol. 2).

[13] Eke, C. I., Norman, A. A., Shuib, L., & Nweke, H. F. (2020). Sarcasm identification in textual data: Systematic review, research challenges and open directions. Artificial Intelligence Review, 53(6), 4215-4258. https://doi.org/10.1007/s10462-019-09791-8

[14] Genc-Nayebi, N., & Abran, A. (2017). A systematic literature review: Opinion mining studies from mobile app store user reviews. Journal of Systems and Software, 125, 207-219. https://doi.org/10.1016/j.jss.2016.11.027

[15] Maynard, D. G., & Greenwood, M. A. (2014). Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In LREC 2014 Proceedings.

[16] Bharti, S. K., Babu, K. S., & Jena, S. K. (2015). Parsing-based sarcasm sentiment recognition in Twitter data. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 1373-1380). https://doi.org/10.1145/2808797.2808910

[17] Veale, T., & Hao, Y. (2010). Detecting ironic intent in creative comparisons. In Proceedings of the ECAI 2010 (pp. 765-770).

[18] Birjali, M., Kasri, M., & BeniHssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowledge-Based Systems, 226, 107134. https://doi.org/10.1016/j.knosys.2021.107134

[19] Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), 5731-5780. https://doi.org/10.1007/s10462-022-10144-1

[20] Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (pp. 174-181). https://doi.org/10.3115/976909.979640

[21] Turney, P. D., & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems, 21(4), 315-346. https://doi.org/10.1145/944012.944013

[22] Han, H., Zhang, J., Yang, J., Shen, Y., & Zhang, Y. (2018). Generate domain-specific sentiment lexicon for review sentiment analysis. Multimedia Tools and Applications, 77(16), 21265-21280. https://doi.org/10.1007/s11042-017-5529-5