

Comparative Analysis Using Machine Learning Techniques for Fine Grain Sentiments

Zeeshan Ahmad¹, Waqas Haider Bangyal¹, Kashif Nisar^{2,3,*}, Muhammad Reazul Haque⁴ and M. Adil Khan⁵

¹Department of Computer Science, University of Gujrat, Gujrat, 50700, Pakistan

²Faculty of Computing and Informatics, University Malaysia Sabah, Jalan UMS, Kota Kinabalu, 88400, Sabah, Malaysia

³Department of Computer Science and Engineering, Hanyang University, Seongdong-gu, Seoul, 04763, South Korea

⁴Faculty of Computing & Informatics, Multimedia University, Cyberjaya, 63100, Selangor, Malaysia

⁵Department Computer Science and Technology, Chang'an University, Xi'an, 710064, China

*Corresponding Author: Kashif Nisar. Email: kashif@ums.edu.my

Received: 21 February 2021; Accepted: 17 May 2021

Abstract: Huge amount of data is being produced every second for microblogs, different content sharing sites, and social networking. Sentimental classification is a tool that is frequently used to identify underlying opinions and sentiments present in the text and classifying them. It is widely used for social media platforms to find user's sentiments about a particular topic or product. Capturing, assembling, and analyzing sentiments has been challenge for researchers. To handle these challenges, we present a comparative sentiment analysis study in which we used the fine-grained Stanford Sentiment Treebank (SST) dataset, based on 215,154 exclusive texts of different lengths that are manually labeled. We present comparative sentiment analysis to solve the fine-grained sentiment classification problem. The proposed approach takes start by pre-processing the data and then apply eight machine-learning algorithms for the sentiment classification namely Support Vector Machine (SVM), Logistic Regression (LR), Neural Networks (NN), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN), Adaboost and Naïve Bayes (NB). On the basis of results obtained the accuracy, precision, recall and F1-score were calculated to draw a comparison between the classification approaches being used.

Keywords: Machine learning; neural network; sentiment analysis system; support vector machine

1 Introduction

In text mining, the 'text' is used to define informative content. All written documents like books, newspapers, and blogs are examples of texts. Nowadays, there is a lot of text available and it is growing day by day. In literature, there are many techniques and approaches available related to text mining



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

for discovering new and useful information from text documents. Text analysis is performed with the help of computer programs and then results are evaluated.

Text mining combines many different fields such as statistics, artificial intelligence, information retrieval, etc., [1]. Text mining is a useful method for examining the informative content of a text or a collection of text. Many text mining approaches are based on the words in the texts. Text mining is a conceptual approach. The concepts in texts help us to understand the current ideas and ideologies of events, objects, subjects, emotions, and actions [2].

In the sentiment analysis, the elements of text are examined to identify whether the text is positive, neutral, or negative i.e., identify the underlying sentiment in the text. Thus, the eigenvalues in the text can be detected. The meanings which are not explicitly mentioned in the text but directed to the objectives can be revealed. In addition, emotional roles in the text can be analyzed [3]. As volume of opinions and comments available on web forums is increasing tremendously with the increase in social media platform users, researchers are strongly interested to make contributions to solve this issue [4]. Sentiment analysis has many different disciplines; there are mainly three approaches to conduct sentiment analysis: one is based on automatic learning, and the other is on the automatic processing of natural language termed as Natural Language Processing (NLP) [5], and the last one combines the first two approaches. Moreover, the sentiment analysis can be conducted on multiple different levels of text such as it can be conducted at document level, sentence level, and word level [6,7]. In document-level sentiment analysis [8], the sentiment of the document depends on the complete document, however, the complete document may contain objective sentences along with subjective sentences that don't contribute to the overall sentiment of the document. The problem of the objective sentences can be solved by performing sentiment analysis at the sentence level, which first separates the subjective sentences from the objective ones and then carries out the sentiment analysis only on subjective sentences [9]. In word-level sentiment analysis, predefined polarity for each word is calculated and the overall sentiment of the phrase or document is computed [10].

With the web advancement and the outbreak of data origins such as blogs, opinion sites, and microblogs, they are necessary to examine the millions of opinions in order to know the thinking of internet surfers. Sentiment analysis is a technology that automatically evaluates the speech, written or spoken, and highlights the various point of view that is expressed by the users related to the specific subject such as a news item, a brand or a product. Multiple approaches of machine learning and evolutionary computing approaches address the task of sentiment analysis [11,12].

In the current era, the need for sentimental analysis is in almost all fields of life including reputation management, politics, marketing, and many more. The most commonly used datasets for sentiment analysis are related to movies and products. The reason behind the utilization of movies' is that the movies are related to different aspects of life, when people talk about any movie, at that time they are actually giving their opinion on different aspects. In this way, a generic dataset is created that can be applied mostly in all domains [13–15].

The remaining part of the study is structured as follows: Literature review is described in Section 2. The presented technique is explained in Section 3. Results are illustrated in Section 4. Conclusion along with future work are discussed in Section 5.

2 Related Work

A lot of work has been done in the area of Sentiment analysis to extract the underlying opinions of people towards any particular subject, trend, product, etc. The available studies focus mainly

on extracting useful information from the natural language and processing it to identify the real sentiments. Sentiment analysis has provided significant benefits in many areas of science and business such as financial and political forecasting, user profiling, and advertising and dialogue systems. Such studies have increased the interest of researchers in Sentiment analysis. However, it is extremely difficult to make sense of thoughts or expressions from multi-mode sources like texts, images, videos, audio recordings, etc. Because it requires an in-depth understanding of the characteristics of an open and covered, regular and irregular source (linguistic, visual, or sound) [16].

Identifying correct underlying sentiments is a challenging task. Multiple classification techniques are implied for the sentiment classification task but there is no hard and fast rule to determine which technique outperforms other techniques. Hence to address this challenging situation [17] conducted a study by making a comparison between various sentiment analysis techniques. The results demonstrated that no classification method outperforms others but multiple various techniques can be merged to overcome the limitations of each other.

Hatzivassiloglou and McKeown in 1997, worked on the document level and used the “World Street Journal” as a data source to develop a Log-Linear Regression model, their work was based on conjunctions and adjectives [18].

Within the same document, Pang et al. in 2002 conducted an analysis with different learning models such as Support Vector Machine (SVM), Maximum Entropy (ME), and Naive Bayes (NB) and also implemented a Bayesian classification with lexicons and documents using posts from blogs, different opinion sites, and political blogs and different film reviews [19].

Multiple researches have been carried out in the literature for carrying out fine-grained classification tasks. One such study was done by [20] by using a deep learning model known as BERT. They have applied this model on the SST dataset and a simple architecture the model outperformed multiple other architectures such as CNN, RNN. A study demonstrated the efficiency of using transfer learning techniques in sentiment classification.

In [21] authors carried out the task of text classification using a recurrent convolutional neural network. The authors applied recurrent structure to find contextual information by employing a max-pooling layer to find key sentiments present in the text. The experiments being conducted on four datasets demonstrated the efficiency of the proposed model.

In [22] authors conducted a comprehensive study with multiple machine learning algorithms for the prediction of sentiments of unseen movie reviews from the Rotten Tomatoes movie review corpus. The study aimed at finding out whether using more fine sentiment annotation during training has an impact on improving accuracy for predicting overall sentiments in the whole test phase.

3 Methodology

In this paper, we used a dataset known as Stanford Sentiment Treebank (SST). The whole SST was introduced by Pang et al. (2005) [21] and it contains of 11,885 total sentences that are actually taken from reviews on movies on rottentomatoes.com. SST is a well-known open-source dataset for the classification of fine-grained sentiment. Additionally, every sentence is analyzed by the rules of Stanford Constituency Parser [22] in the form of a tree, where the complete sentence is taken as root and single words are considered as a leaf. Entirely, SST includes 215,154 distinctive phrases of various sizes each phrase was assigned a sentiment by 3 judges. This way the dataset was converted into a 5-class classification dataset (SST-5) known as a fine-grained sentiment analysis dataset. Each class

has been assigned a number from 0–5. This dataset is embedded to train the models for learning the sentiment of sentences, phrases, and words together.

We also managed to convert this data into a 2-class classification dataset (SST-2) called binary classification sentiment analysis dataset. We discarded neutral sentences, converted negative and partially negative to negative, and converted partially positive and positive to positive. This way we converted the dataset to SST-2 losing neutral sentences.

Fig. 1 represents the detailed system architecture being implemented to carry out this study. Further, we have used all nodes and only roots for both of the datasets SST-2 and SST-5 (for the classification of all five classes). In All nodes version of the dataset all phrases are used, while in the root version we only use root nodes of the sentiment tree or the complete sentence. The sentence length distribution for SST-2 and SST-5 can be seen in Figs. 2 and 3 respectively.

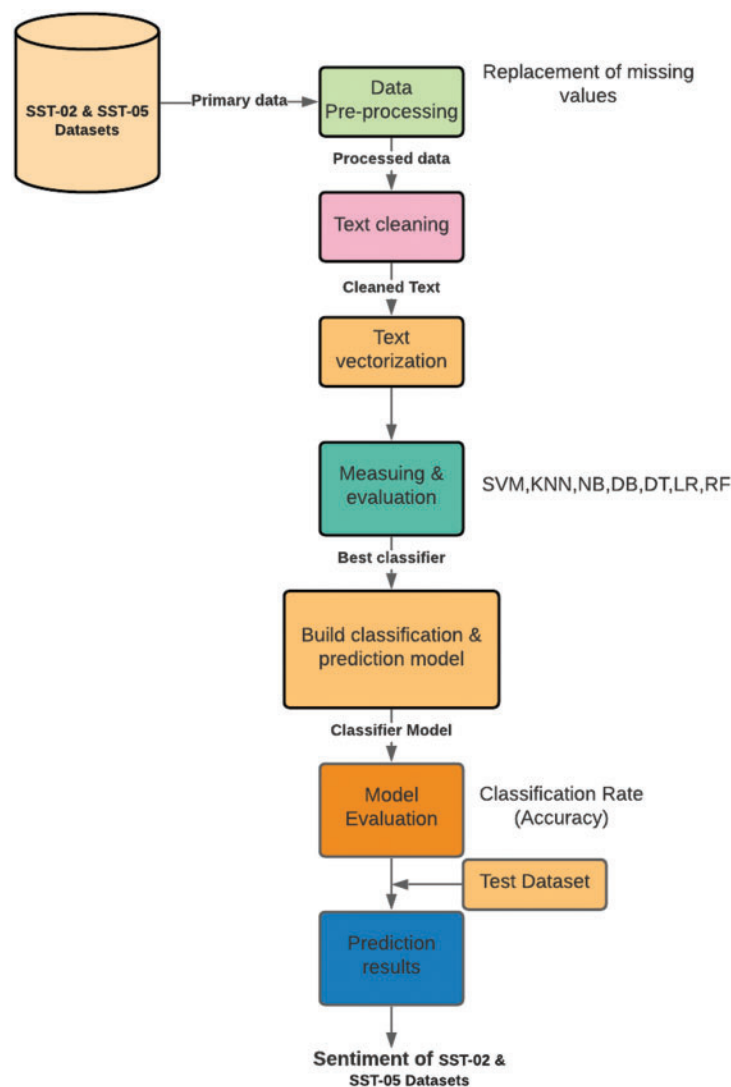


Figure 1: System architecture

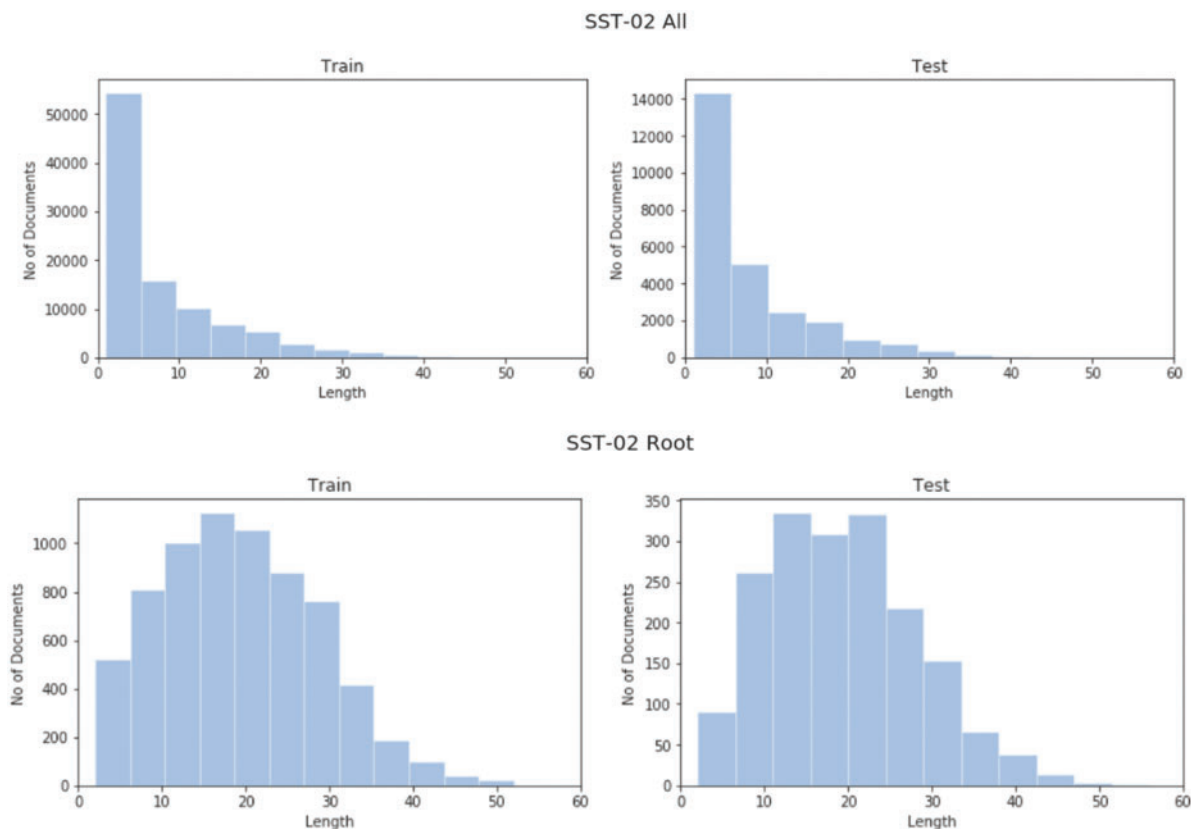


Figure 2: Sentence length's distribution of SST-02 (All, Root)

3.1 Data Set

In our study, we used a dataset known as Stanford Sentiment Treebank which consist of movie reviews from Rotten Tomatoes. The dataset originally is derived from the reviews dataset used by Pang et al. (2005) [21] along with the corpus comprises 11,855 unique movie review sentences. Later on, it was parsed with the help Stanford Parser [23] that derived 215,154 unique phrases and then it was annotated by Amazon Me-chanical Turk. The most appeared word tokens in training and testing sets can be seen in Fig. 4.

The methodological approach that is being followed can be briefed in following four given steps:

3.1.1 Data Pre-processing

Data is thoroughly cleaned during the pre-processing process, during this particular phase, we applied a number of cleanings along with bag-of-words technique with the TF-IDF method for finding out the score of every word. In our study, we split the sentence trees into phrases and words. In this way, the classifier can better distinguish the unseen data of the real world, because it will be trained on the short phrases consisting of even a single word up to the complete sentences. In another split, we only use the root sentences that were originally in the dataset of Pang et al. (2005) [21]. We performed the following basic text pre-processing techniques in our work.

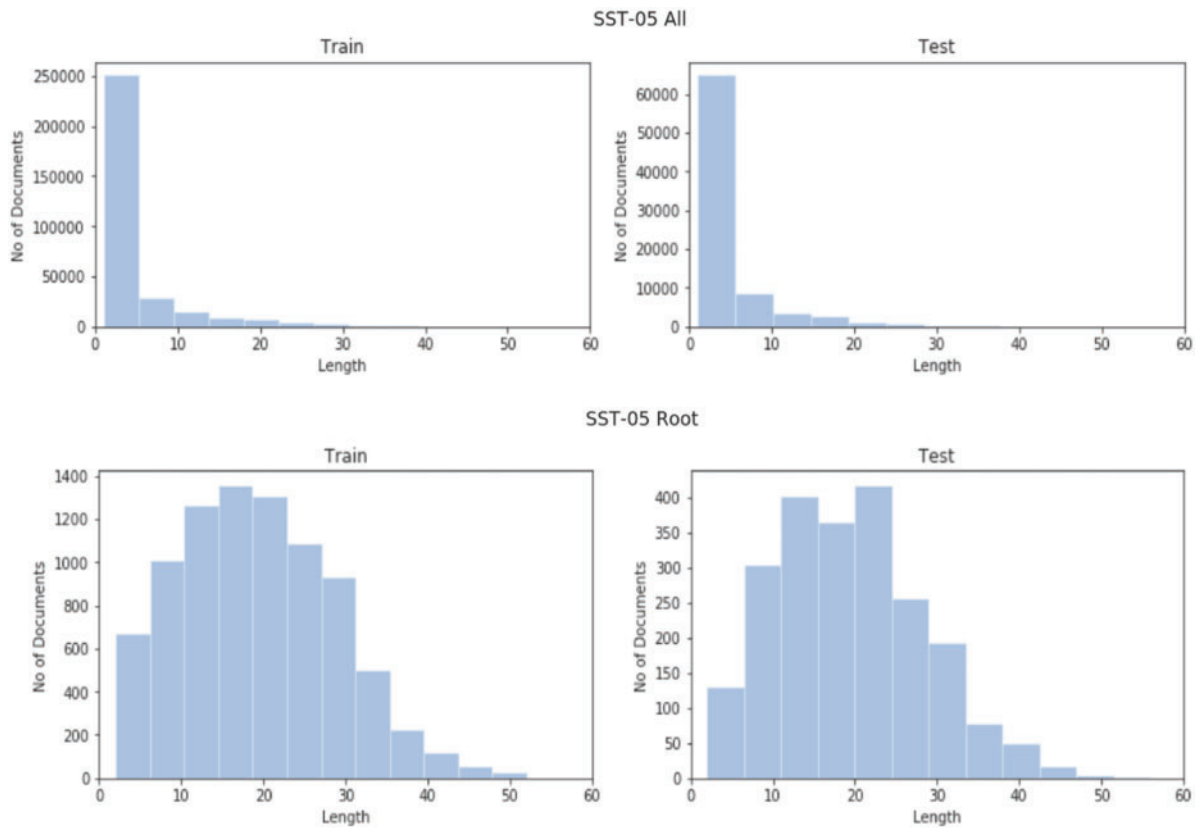


Figure 3: Sentence length’s distribution of SST-05 (All, Root)



Figure 4: Most common words in training and testing sets

- **Converting into Lowercase:** We converted the corpus into lowercase.
- **Tokenization:** We tokenize the sentences and phrases. This is usually done to split long sentences into small meaningful tokens. This step is also considered the text segmentation phase.
- **Lemmatize:** this phase is related to stemming different words into their base form. So using the step we bring a word to its base form. For instance, convert “enjoying” to its lemma i.e., “enjoy”.

3.1.2 Classification Model Building

We choose the following most competent classification algorithm based on the results, the details of classifiers being used is given as follows:

Support Vector Machine (SVM)—a well-known Machine Learning classifier is known as SVM is also applied in this study. It is been applied to both linear along with non-linear problems that has demonstrated visible results in different real-life applications [24]. SVM works by separating the data into classes by using a hyper-plane. It works great with non-linear real world problems due to the presence of kernels that takes low dimensional input space and then convert it to high dimensional space. We can say that SVM is capable of doing complex data transformations and moving data to different classes.

Logistic Regression (LR)—Among the most used ML classifiers is LR. A statistics based model that actually uses variable vectors and find out the variable weights and thus helps predicting the variable class in the word vector form. LR can only be used only if the dependent variable is only binary. No linear relationship exists between dependent and independent variable in LR and moreover the independent variable could not be normally distributed [25].

Naïve Bayes—A probabilistic classifier that is based upon the Bayes Theorem. The basic reason behind its popularity is its accuracy, simplicity, and reliability. Being applied to multiple real-life applications, but has its applications in natural language processing problems. The main assumptions is that each feature try out equal and independent contribution to the main outcome. It actually calculates the attribute probability using preceding piece of information related to the attribute.

Adaboost—Adaboost is called Adaptive Boosting it happens to be the first Boosting Algorithm to be known in the machine learning. For the sole purpose of converting lazy learners into eager learner Boosting algorithms are most commonly used [26]. Adaboost is commonly used to enhance the lazy learner’s prediction capability. It works by combining several slow learners to make one strong learner and works iteratively. Primarily, all instances have similar weights and in next iterations misclassified instances weights are updated correctly classified instances weights are decreased and weights of misclassified instances increases. All the instances are assigned similar weights and in the next iterations, weights of misclassified instances.

K-NN—*KNN* classifies the instances to the nearest neighbor that has majority vote. To find out the most near neighbor classifier make use of distance metric and training instances and find out instances with least distance. The distance is taken between the test instances and all the training instances. The distance is measured using Euclidean distance [27]. The feature value is calculated of nearest neighbor training examples and majority of it is taken as prediction value and further new dataset is categorized. It tends to give highly exact predictions thus it is used for the applications that require high accuracy.

Decision Trees—Decision trees represent decisions in the form of a tree where leaves represent class attributes and inner nodes shows attributes in descriptive form. They have most widely used applications in data mining. Sometimes made upside down with root at the top. Interpretation is quite

easy therefore they are most commonly used. For some given node X, it's all children matches all the values that could be possible associated attributes. Highly robust to noisy data. The KNN works by finding out the best characteristics that gives maximum classification process information. The whole process finally stops when all leaf node become pure as whole i.e., all the instances are classified or no further classification required [28].

Random Forest (RF)—RF a well-known classifier was proposed by Breiman and Cutler. An ensemble method that works by binding the subspaces and bagging [29]. It actually builds a set of decision trees from the already present training dataset [30,31]. The label of whose is decided after getting votes from multiple decision trees. It is used to accurately classify large datasets. Its applications include remote sensing, drug discovery, remote sensing, network intrusion detection, etc. . . .

Multi-Layer Perceptron—MLP belongs to the family of feed-forward artificial neural networks (ANN). ANN works by mimicking the human brain behavior. Just the same way human receives input grasps it, and then further generates responses is the main stimulation behind ANN. ANN learn training data representation and then relate it with the desired output variable. Perceptron being basic unit of ANN. Each perceptron use some weighted input and then produces output using some activation function. They are widely used in multiple real-world applications such as Character Recognition, Protein Secondary Structure, Data Compression, Computer Vision, Pattern Recognition, and Speech Recognition.

3.1.3 Model Evaluation and Testing

Training and evaluation of different classifiers are done by using several performance measures such as accuracy measure, precision, recall, F1-Score, etc. Then the evaluation of test data that consists up of unclassified views, was conducted to predict the sentiment class of each review among this set.

3.1.4 Data Vectorization

This step involves the transformation of simple text to digital vectors because most learning algorithms do not take text in a direct form but work using digital vectors. To address this purpose we performed a transformation of text to digital vectors.

4 Experiment and Results

Data pre-processing is done according to the following steps: Replace missing values, removing noise or Selection, Pre-processing text (Transform cases, Tokenization, Filtering Tokens, Filtering Tokens, Filter stop words, Stemming).

In the data vectorization step, it's no matter which classification method is used, the first step is to represent the documents so that they can be automatically processed by the classifiers. Most approaches are based on the vector representation of documents, here we have used TF-IDF coding which gives a view to documents in the form of rows and terms in the form of columns. The IDF function is calculated as follows: $\log[(1+n)/(1+df(d,+))]+1$.

Here, the variable N indicates the total count of documents in the document library. The Term Frequency Inverse Document Frequency (TF-IDF) score for a function is calculated as follows:

$$TD - IDF = TF * IDF \quad (1)$$

For evaluating multiple classifiers listed above we used precision, recall and F1 score and accuracy measures to look out the most effective model that has the highest measurement values, a standard

technique for evaluation known as cross-validation, a systematic way used to perform repeated percentage splits, that works by dividing the data set into 10 folds and then further takes each piece, in turn, to test it and counts the remaining 9 pieces as training data, 9 subsets of these data are used as learning data set to make a model and the remaining subsets are further applied for verification model. This gives total of 10 evaluation results the average of these results is taken after running dataset with above mentioned eight machine learning algorithms. To prepare for the automatic model, we created a reverse document frequency vectorization called (TF-IDF) term. The result obtained as a result of this vectorization is a hollow matrix that has a representation of each different sentence as a vector, the vector is said to have the same length as our vocabulary, i.e., list in our observed data, each word representing an entry in the vector.

Experimental results demonstrated in [Tab. 1](#) shows that on SST-02, logistic regression and neural network model reached a high accuracy (85%), compared to the other models. The results for SST-05 are elaborated in [Tab. 2](#) which shows that Neural Network, model reached a high accuracy (85%), compared to the other models.

Table 1: Accuracy, precision, recall and F1 score for classification algorithms on SST-02 dataset

Model	All				Root			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Logistic regression	0.8555	0.8506	0.8586	0.8465	0.8034	0.8032	0.8046	0.8035
Random forest	0.7918	0.7811	0.7993	0.7761	0.7161	0.7135	0.7244	0.7162
Decision tree	0.7400	0.7303	0.7376	0.7276	0.6139	0.6139	0.6140	0.6140
SVM	0.6948	0.6803	0.6913	0.6784	0.8155	0.8154	0.8160	0.8155
KNN	0.7385	0.7219	0.7452	0.7186	0.7748	0.7740	0.7794	0.7750
Adaboost	0.7283	0.6982	0.7581	0.6984	0.7282	0.7271	0.7320	0.7283
MLP/BPA	0.8545	0.8499	0.8563	0.8464	0.8138	0.8137	0.8146	0.8139
Naïve Bayes	0.8305	0.8262	0.8289	0.8243	0.8210	0.8209	0.8212	0.8210

Table 2: Accuracy, Precision, Recall and F1 score for classification algorithms on SST-05 dataset

Model	All				Root			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Logistic regression	0.7655	0.4983	0.6473	0.4414	0.4100	0.3526	0.4429	0.3585
Random forest	0.7353	0.3910	0.6805	0.3406	0.3543	0.2013	0.5166	0.2809
Decision tree	0.7370	0.4728	0.5110	0.4498	0.2765	0.2635	0.2695	0.2627
SVM	0.6567	0.3703	0.5075	0.3464	0.3946	0.2748	0.4450	0.3205
KNN	0.7490	0.4769	0.6387	0.4159	0.3706	0.3554	0.3932	0.3459
Adaboost	0.7194	0.3832	0.5429	0.3410	0.3529	0.3060	0.3609	0.3109
MLP/BPA	0.7893	0.5796	0.6450	0.5409	0.4127	0.2986	0.4297	0.3392
Naïve Bayes	0.7295	0.3909	0.5092	0.3531	0.4104	0.3719	0.3935	0.3720

The results are presented in graphics form to visualize this performance evaluation between the models built with the different classification algorithms on the dataset [Fig. 5](#) shows the results for

SST-02 and Fig. 6 for SST-05. The Performance operator is used to evaluate the performance of a classifier. It provides a list of performance criteria values.

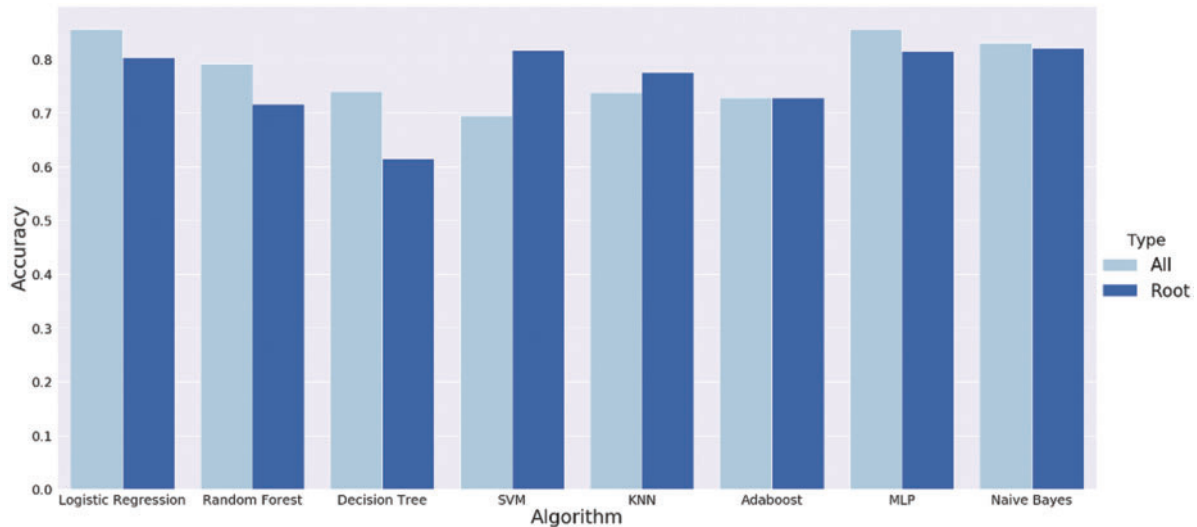


Figure 5: Graphical representation of accuracy result scores for SST-02 dataset

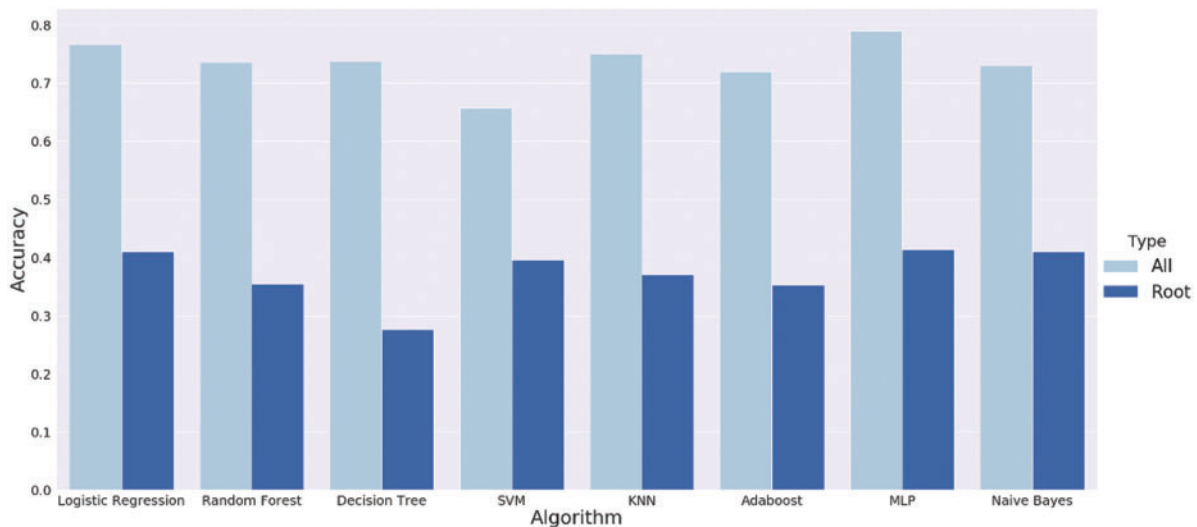


Figure 6: Graphical representation of accuracy result scores for SST-05 dataset

On the grounds of accuracy for further analysis, we can discover few more parameters such as precision, F-score, and recall.

Precision is the fraction of analogous occurrences getting involved in the retrieved occurrences. Recall-also called sensitivity, recall is the fraction of complementary occurrences that have been retrieved from the overall occurrences. Both recall and precision are adequately based on the understanding and design of relevance. F-score-is a measure that integrates the recall and precision, it is basically the harmonic mean of recall and precision.

5 Conclusion

In this paper, we mainly address the comparative analysis of eight different machine learning algorithms namely Random-Forest, Support Vector Machine, K-Nearest Neighbor, Naïve Bayes, Logistic Regression, Decision Tree, Adaboost, and MLP that can be used for the classification of fine-grained sentiment on the SST datasets. A comparative analysis of these machine learning approaches shows that these algorithms are simpler and efficient. Neural Network (MLP) and logistic regression with the best accuracy can be considered as a benchmark for all other algorithms. Furthermore, the system has been analyzed in the aspects of precision, recall, and F1-score for all the algorithms.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. Dang and P. H. Ahmad, "Text mining: Techniques and its application," *International Journal of Engineering & Technology Innovations*, vol. 1, no. 4, pp. 22–25, 2014.
- [2] M. Song and Y. B. Wu, *Handbook of research on text and web mining technologies*. Hershey, Pa.: IGI Global, 2010.
- [3] A. Joshi, P. Bhattacharyya and S. Ahire, "Sentiment resources: Lexicons and datasets," in *A Practical Guide to Sentiment Analysis*, vol. 5, Cham: Springer, pp. 85–106, 2017.
- [4] S. Lei, S. Bai, K. Liang and Z. Yan, "Web forum sentiment analysis based on topics," in *Proc. IEEE 9th Int. Conf. on Computer and Information Technology*, Xiamen, China, pp. 148–153, 2009.
- [5] A. P. Gopi, R. N. S. Jyothi, V. L. Narayana and K. S. Sandeep, "Classification of tweets data based on polarity using improved RBF kernel of SVM," *International Journal of Information Technology*, 2020. <https://doi.org/10.1007/s41870-019-00409-4>.
- [6] A. Kumar and M. S. Teeja, "Sentiment analysis: A perspective on its past, present and future," *International Journal of Intelligent Systems and Applications*, vol. 4, no. 10, pp. 1, 2012.
- [7] W. H. Bangyal, J. Ahmad, H. T. Rauf and R. Shakir, "Evolving artificial neural networks using opposition based particle swarm optimization neural network for data classification," in *2018 Int. Conf. on Innovation and Intelligence for Informatics, Computing, and Technologies*, Sakhier, Bahrain, pp. 1–6, 2018.
- [8] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *ACL '05: Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor Michigan, pp. 115–124, 2005.
- [9] S. M. Kim and E. Hovy, "Determining the sentiment of opinions," in *COLING 2004: Proc. of the 20th Int. Conf. on Computational Linguistics*, Geneva Switzerland, pp. 1367–1373, 2004.
- [10] W. H. Bangyal, J. Ahmad, H. T. Rauf and S. Pervaiz, "An overview of mutation strategies in bat algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 8, pp. 523–534, 2018.
- [11] M. Junaid, W. H. Bangyal and J. Ahmad, "A novel bat algorithm using sobol sequence for the initialization of population," in *2020 IEEE 23rd Int. Multitopic Conf. (INMIC)*, Bahawalpur, Pakistan, pp. 2–7, 2020.
- [12] S. M. Vohra and J. B. Teraiya, "A comparative study of sentiment analysis techniques," *Journal of Information, Knowledge and Research in Computer Engineering*, vol. 2, no. 2, pp. 313–317, 2014.
- [13] M. Dragoni, S. Poria and E. Cambria, "Ontosentinet: A commonsense ontology for sentiment analysis," *IEEE Intelligent Systems*, vol. 33, no. 3, pp. 77–85, 2018.
- [14] W. H. Bangyal, A. Hameed, W. Alosaimi and H. Alyami, "A new initialization approach in particle swarm optimization for global optimization problems," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 6628889:1–17, 2021.

- [15] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, "A systematic literature review on particle swarm optimization techniques for medical diseases detection," *Computational and Mathematical Methods in Medicine*, vol. 2021, pp. 5990999:1–10, 2021.
- [16] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.
- [17] P. Duan, G. Mao, C. Zhang and B. Zhang, "Applying DCOP to user association problem in heterogeneous networks with markov chain based algorithm," arXiv preprint arXiv:1701.01289, 2017.
- [18] M. Munikar, S. Shakya and A. Shrestha, "Fine-grained sentiment classification using BERT," in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, Kathmandu, Nepal, pp. 1–5, 2019.
- [19] J. Sun, R. Jin, X. Ma, J. young Park, K. ah Sohn *et al.*, "Gated convolutional neural networks for text classification," *Advances in Computer Science and Ubiquitous Computing*, vol. 715, pp. 309–316, 2021.
- [20] J. Y. Wu and Y. Pao, "Predicting sentiment from rotten tomatoes movie reviews reviews," 2012. [Online]. Available: https://nlp.stanford.edu/courses/cs224n/2012/reports/WuJean_PaoYuanyuan_224nReport.pdf.
- [21] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *ACL-05-43rd Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, Ann Arbor Michigan, vol. 3, no. 1, pp. 115–124, 2005.
- [22] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, Seattle, USA, pp. 1631–1642, 2013.
- [23] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks," in *EMNLP*, Doha, Qatar, pp. 740–750, 2014.
- [24] K. P. Bennett and C. Campbell, "Support vector machines: Hype or hallelujah?," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 1–13, 2000.
- [25] A. Prabhat and V. Khullar, "Sentiment classification on big data using naïve Bayes and logistic regression," in *2017 Int. Conf. on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2017.
- [26] E. Ahmed, "Reviews using machine learning," in *2019 10th Int. Conf. Comput. Commun. Netw. Technol.*, New York, USA, pp. 86–91, 2019.
- [27] B. G. Priya, "Emoji based sentiment analysis using KNN," *International Journal of Scientific Research and Review*, vol. 7, no. 4, pp. 859–865, 2019.
- [28] A. Suresh and C. R. Bharathi, "Sentiment classification using decision tree based feature selection," *International Journal of Control Theory and Application*, vol. 9, no. 36, pp. 419–425, 2017.
- [29] Ankit and N. Saleena, "An ensemble classification system for twitter sentiment analysis," *Procedia Computer Science*, vol. 132, pp. 937–946, 2018.
- [30] A. Ashraf, S. Pervaiz, W. H. Bangyal, K. Nisar, A. Ag. Ibrahim *et al.*, "Studying the impact of initialization for population-based algorithms with low-discrepancy sequences," *Applied Sciences*, vol. 11, no. 17, pp. 8190, 2021.
- [31] W. H. Bangyal, K. Nisar, A. Ag. Ibrahim, M. R. Haque, J. Rodrigues *et al.*, "Comparative analysis of low discrepancy sequence-based initialization approaches using population-based algorithms for solving the global optimization problems," *Applied Sciences*, vol. 11, no. 16, pp. 7591, 2021.