TeenSenti - A novel approach for sentiment analysis of short words and slangs

Sahil Kamath
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai, India
sahilkamath0108@gmail.com

Vaishnavi Padiya
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai, India
vaishnavipadiya@gmail.com

Sonia D'Silva
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai, India
sonia.dsilva.2003@gmail.com

Nilesh Patil
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai, India
nilesh.p@djsce.ac.in

Meera Narvekar
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai, India
meera.narvekar@djsce.ac.in

***Abstract*—The present prevalence of online platforms and the internet has seen a substantial increase in the utilization of Generation Z slang, abbreviated expressions, and short words. These Lexical subtleties have become a vital part of a person's daily interactions, and therefore evident in reviews, product comments, and throughout the internet. Resultantly, there exists a need to integrate these Informal expressions such as abbreviations, slang, and short words into natural language processing (NLP) systems. The Informal expressions naturally contain some contextual relevance and therefore can be used to improvise sentiment analysis, as these expressions contribute significantly to the overall context and sentiment of sentences. However, traditional NLP techniques often fall short in recognizing and incorporating these informal expressions, resulting in an attenuated accuracy of sentiment analysis. To address this issue, this study ventures to provide a comprehensive slang dictionary, encompassing short words, abbreviated expressions, and slang along with the sentiment and sentences of each. This curated slang dictionary is effectively integrated into NLP systems using FastText Embeddings. Through extensive testing across multiple machine learning (ML) and deep learning (DL) models, this approach significantly enhances the accuracy of sentiment analysis when compared to conventional methodologies. This research emphasizes the importance of accommodating informal expression in NLP systems thus opening the possibility of future research.**

***Index Terms*—Sentiment Analysis, Slang, Short Words, NLP, FastText Embeddings**

I. Introduction

In the digital era, characterized by the prevalent use of brief and informal expressions, the impact of such linguistic patterns stretches beyond the boundaries of social media, incorporating the entire internet. Abbreviations, like "osm" signifying "Awesome" and "DNW" standing for "do not want" have become ubiquitous, encroaching on various online platforms, including product reviews and comment sections. These linguistic subtleties entail sentiments that may be positive or negative. Considering the examples stated, osm signifies enthusiasm, approval, admiration, or it describes something that is, remarkable, impressive, or exceptionally good therefore indicating an overall positive sentiment, similarly "DNW", implies a lack of desire, disapproval, or unwillingness towards something resulting all in all a negative sentiment. In light of this, the linguistic nuances play a vital role in deciding the total sentiment of the text so, a careful consideration of these linguistic subtleties becomes indispensable when engaging in sentiment analysis endeavors. The ongoing and dynamic transformation of the internet introduces a perpetual expansion of the vocabulary of slang and abbreviated expressions. These changes impose a limitation on the existing online datasets due to the absence of modern linguistic nuances and also prove inadequate for conducting sentiment analysis and other NLP tasks. In addition to this, the existing datasets lacked the inclusion of recent slang and abbreviations and their associated sentiments. Acknowledging this evident disparity, a need emerges to construct a new dataset that holds the recent slang along with their possessed sentiments. The essence of this research is the overarching objective of enhancing the accuracy of NLP models. The following research was then started to overcome the limitations in the existing systems and preceded by curating a dataset consisting of 20,000 sentences containing slang words, short words, and abbreviations along with their inherited sentiments and sentences for it. The subsequent sections of the paper demonstrate the overall process.

II. Literature Review

Sentiments refer to the emotions, opinions, or attitudes expressed in a piece of text. In Natural Language Processing (NLP), sentiment analysis is crucial for understanding and extracting sentiments from textual data. It helps determine whether the expressed sentiment is positive, negative, or neutral, providing valuable insights for various applications such as customer satisfaction, brand reputation, political analysis, and financial industry applications[1]. Even before the upscaling of social media S. Kiritchenko, Svetlana, Xiaodan Zhu,

and Saif M. Mohammad [2] performed sentiment analysis for short texts, excelling in tweet and SMS sentiment detection, by utilizing a supervised statistical approach with diverse features and tweet-specific sentiment lexicons. The need for sentiment analysis in NLP arises from the vast amount of unstructured text data available online. As people increasingly communicate through digital platforms, there is a growing volume of user-generated content in the form of reviews, social media posts, and comments. Analyzing sentiments allows businesses and researchers to gauge public opinion, make informed decisions, and enhance user experiences. Jayant Mishra[3] used Supervised Machine Learning techniques to perform sentiment analysis on a labeled Twitter Dataset available on Kaggle for classification. The rise of social media prevailed the increase in the usage of slang, short words, and informal language on the internet, which is driven by factors such as character limits in social media, the desire for quick communication, and the influence of online communities. Slangs and informal language can be more expressive and efficient for conveying emotions or ideas within the constraints of these platforms. Additionally, the internet fosters a sense of informality and a desire for authenticity, leading to the widespread adoption of casual language in online communication. NLP techniques, including sentiment analysis, adapt to these linguistic trends to accurately interpret and understand the sentiment behind informal expressions. Sakhawia Farogh[4] emphasized the importance of developing advanced techniques to enhance the accuracy and context-awareness of sentiment analysis models. Harshali Patil and Mohammad Atique [5] highlighted the importance of social media as a valuable platform for tracking and exploring public sentiment, as millions of users share their feelings and opinions on platforms like Twitter and Facebook. It emphasizes the need for further research in handling challenges such as negations, hidden sentiments identification, slang, and polysemy. Their research concludes that sentiment analysis is a crucial area of research, given the abundance of user-generated content on social media platforms, and highlights the need for more advanced techniques to analyze and classify sentiment effectively. A framework that incorporates slang words and emoticons to improve the performance of sentiment analysis models on social media text was proposed by Paramita Dey and Soumya Dey [6]. They found that incorporating slang words and emoticons led to an improvement in the accuracy of sentiment analysis models. Vivank Sharma, Shobhit Srivastava, B. Valarmathi & N. Srinivasa Gupta [7] employed a deep learning approach, utilizing a snowball stemmer and three algorithms (LSTM, LR, CNN) for sentiment analysis of slang in tweets, with the LSTM network. Shelley Gupta, Shubhangi Bisht & Shirin Gupta [8] performed sentiment analysis on online user's opinions about a product or service from Twitter and other Social Media datasets by using a lexicon-based approach that incorporates text and slang. It also evaluates the sentiment polarity of tweets using machine-learning classification techniques. Kundeti Naga Prasanthi, Rallabandi Eswari Madhavi, Degala Naga Sai Sabarinadh & Battula Sravani [9] proposed a sentiment analysis model for Twitter using BERT and RoBERTa transformer models. It highlights the advantages of using these transformers, such as their ability to capture contextual information, understand relationships between words, and handle informal language used in tweets. K. Manuel, Kishore Varma Indukuri & P. Radha Krishna [10] introduced a method for sentiment mining in online communities, specifically addressing slang words, using Delta Term Frequency and Weighted Inverse Document Frequency technique, providing sample results and emphasizing the development of an approach for handling Internet slang in sentiment analysis.

III. Methodology

A. Dataset Collection

The research began by gathering a comprehensive collection of informal expressions evident on the internet such as the internet slang, short words, and abbreviations which captured a wide range of colloquial expressions vital in Internet communication. Using this data collection, a massive dataset of approximately 20,000 sentences was generated. These sentences were thoroughly crafted with the help of LLM to include sentences for both positive and negative sentiments of each slang word. LLMs were chosen to generate a legally compliant dataset which also included controlled integration of slang and short words which enhanced the model's proficiency in understanding informal expressions. Each sentence was structured and generated in a way such that the context of the sentence was affected by the context of the slang word, therefore highlighting the fact that internet slang and short words contribute to the overall sentiment of a text. This deliberate construction aimed to ensure clarity and precise interpretation.

B. Custom Tokenization for Enhanced Context Preservation

The comprehensive dataset then went through the preprocessing pipeline, utilized a customized tokenizer to overcome the limitations faced by the standard and default tools of the NLTK library. The conventional approach could not effectively recognize the short words and slang words therefore it discarded them. To overcome this, the custom tokenizer scrutinized through all the words and discarded only those short words which were absent in authors' curated slang dictionary. This strategy ensured the preservation of short words and internet slang, contributing to a more subtle understanding of the context and therefore improved sentiment analysis.

C. Train and Test Dataset Generation for Mitigating Overfitting

To prevent overfitting, the dataset was partitioned into training and testing datasets such that 80% of sentences associated with every acronym or slang were allocated to the training dataset, while the rest 20% were allocated to the testing set, this method was repeated throughout all sentences containing various slang phrases and short words in the dataset, ensuing distinct training and testing datasets for each acronym or slang. Cohesive training and testing datasets were then constructed by combining the individual training and testing datasets.

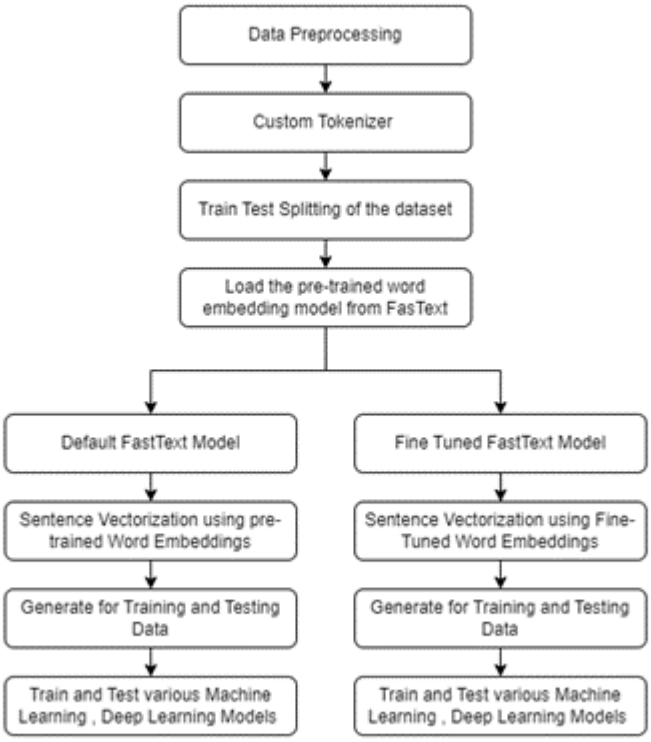

Fig. 1. Model Architecture.

D. FastText Embedding

To capture subword records and take care of out-of-vocabulary words correctly a FastText Embedding model was used to encode the internet slang dataset into vectors. To do this a pre-trained FastText word embedding model wiki-news-300d-1M-subword[11], which is trained on one million word vectors from Wikipedia 2017 data was loaded using pickle. This helped safeguard the linguistic capabilities inherent inside the internet slags and brief phrases, which ended in an enhancement of the version's ability to apprehend the casual expressions with their intended meanings.

E. Model Comparison and Fine-Tuning

The FastText word embedding model was then satisfactorily tuned on authors' dataset. To test the Fine-Tuned version 2 sets of training and testing datasets had been generated. The first set incorporated the embeddings from the Non-Fine-Tuned FastText model, while the second set contained embeddings from Fine-Tuned FastText model. The generated training datasets were rigorously implemented throughout a huge variety of Models, starting from Machine Learning, Deep Learning, and Natural Language Processing Models. Each model's performance, particularly in sentiment analysis, was critically evaluated in opposition to baseline results. This stage aimed to determine how incorporating informal expressions affected the accuracy of various fashions and their understanding of sentiments in Internet communication.

F. Evaluation Metrics and Result Analysis

Quantitative evaluation criteria were used to evaluate the effectiveness of fine-tuned model before and after including authors' dataset with more slang. Various metrics including accuracy, precision, and f1 score were evaluated to capture the skill of the model in perceiving contexts with lots of informal texts. The methods adopted in this study required internet short words and slang and informal context integrated into NLP models. Through comprehensive dataset curation, refinement of the FastText embedding model, and comparative analysis through quantitative evaluation criteria, this approach aims to provide an understanding of NLP processes in the context of Internet slang and non-linguistic features.

G. Deploying the model on Hugging Face

The model was deployed on Hugging Face[12], a popular platform where natural language processing models are hosted and shared. Using the Hugging Face API, it was shown that the model performed well with respect to some given dataset. Also, fine-tuned Roberta model was also better than [13] on analysing sentences with slangs, which is available on Hugging Face as shown by comparative analysis. The comparison statistics can be found in Table IV

IV. Results

The Fine-Tuned Word Embedding Model was diffused across diverse machine learning and deep learning models to assess the quality of its performance in relation to the Non-Fine-Tuned one.

a) Logistic Regression: It analyzes the relationship between two factors and predicts the value of one of the factors based on the relationship observed. This is a kind of probabilistic classifier that relies on supervised machine learning. It applies a linear model with a logistic function to estimate the probability of sentiment classes. The Logistic Function is written as

$$F(x) = 1/1 + e^{-x} \tag{1}$$

On using the embeddings generated by the Fine-Tuned fastText word Embedding model the accuracy obtained was 0.87, whereas the accuracy obtained by using the Non-Fine-Tuned fastText model is 0.78. Resulting in a better accuracy of the Fine-Tuned Model. For PR Curve - The Precision-Recall curve for the Fine-Tuned model exhibited a more pronounced inclination towards the top-right corner compared to the curve for the Non-Fine-Tuned model. This observation indicated that the Fine-Tuned model was a superior classifier than the Non-Fine-Tuned model, suggesting that it made more accurate positive predictions while capturing a higher proportion of actual positive instances. The AUC-PR for the Non-Fine-Tuned version was 0.64, which was increased to 0.77 for the Fine-Tuned model. This discrepancy in AUC-PR values

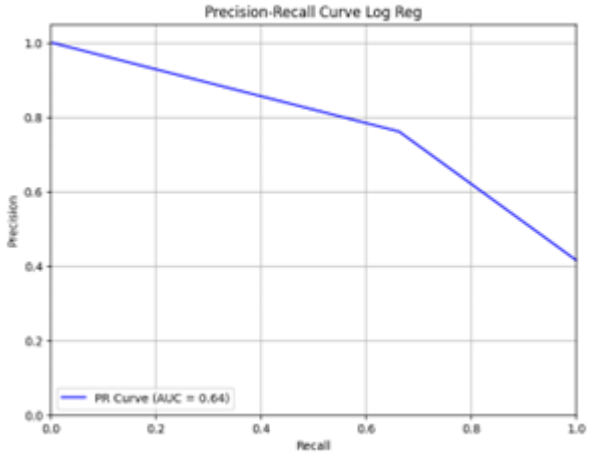

Fig. 2. Logistic Regression PR Curve for Non-Fine-Tuned Model.

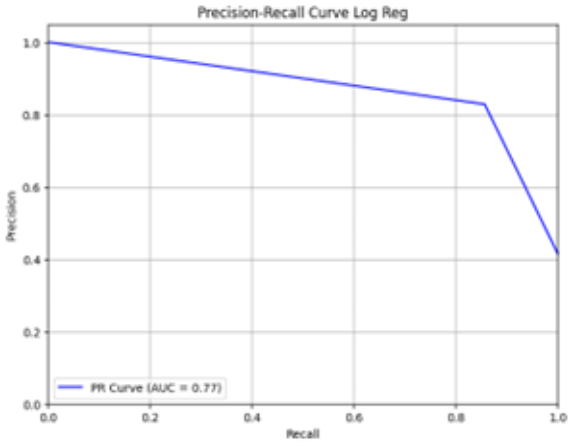

Fig. 3. Logistic Regression PR Curve for Fine-Tuned Model.

similarly supported the realization that the Fine-Tuned version outperformed the Non-Fine-Tuned version in terms of precision and recall.

b) Decision Tree Classifier: Decision Tree Classifier is a versatile algorithm used for both classification and regression that makes decisions by recursively splitting the dataset based on features, creating a tree-like structure. It employs a tree-like structure to make hierarchical decisions and categorize text sentiments. The formulas for Entropy and Gini Index are as follows, they help in deciding the best splitting for the decision tree.

$$\text{Entropy} \ = \sum_{i=1}^{C} -p_i * \log_2 (p_i) \tag{2}$$

$$\text{Gini} \ = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{3}$$

where C is the number of classes. On using the embeddings generated by the Non-Fine-Tuned fastText word Embedding model the accuracy obtained was 0.71, whereas the accuracy obtained by using the Fine-Tuned fastText model is 0.76. Resulting in a better accuracy of the Fine-Tuned Model.

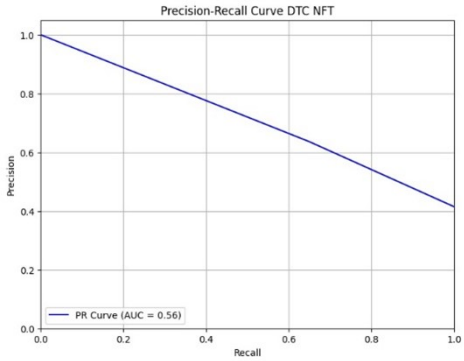

Fig. 4. DTC PR Curve for Non-Fine-Tuned Model.

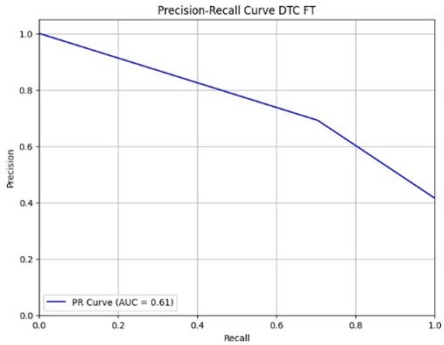

Fig. 5. DTC PR Curve for Fine-Tuned Model.

The Fine-Tuned PR curve is positioned closer to the top right compared to the Non-Fine-Tuned PR curve, signaling the superior classification ability of the Fine-Tuned model. This suggested that the Fine-Tuned model excelled in making accurate positive predictions while capturing a greater chunk of actual positive instances. Additionally, the AUC-PR of the Fine-Tuned model, equal to 0.61, outperformed the Non-Fine-Tuned AUC-PR curve by 0.05 (0.56). This indicated that the Fine-Tuned model demonstrates better overall performance in terms of precision and recall.

c) K-Nearest Neighbours(KNN): It is an algorithm that is used to classify data points based on the majority class in which the k nearest neighbors belong. It classifies sentiments based on the majority class of k nearest neighbors in sentiment analysis. The formula for KNN is - Euclidean Distance Calculation:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4}$$

where $d(x, y)$ is the Euclidean distance between points x and y.

On using the embeddings generated by the Non-Fine-Tuned fastText word Embedding model the accuracy obtained was 0.83, whereas the accuracy obtained by using the Fine-Tuned

fastText model was 0.89. Resulting in a better accuracy of the Fine-Tuned Model.

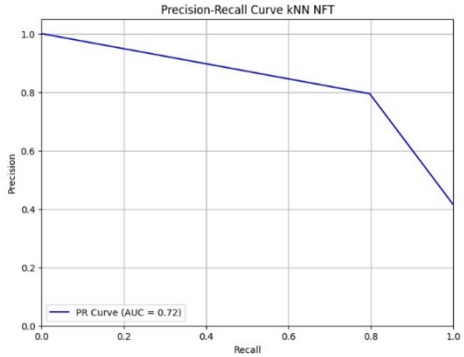

Fig. 6. KNN PR Curve for Non-Fine-Tuned Model.

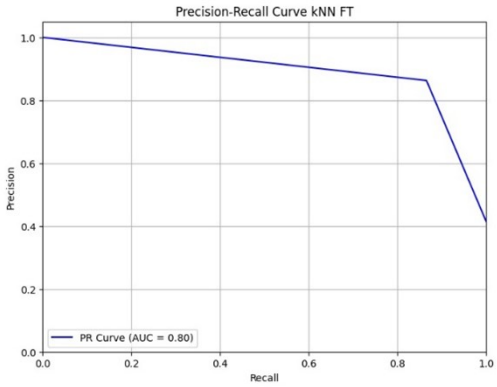

Fig. 7. KNN PR Curve for Fine-Tuned Model.

The Fine-Tuned PR curve was closer to the top right, indicating the superior classification ability of the Fine-Tuned model than the Non-Fine-Tuned Precision-Recall curve. This implied that the Fine-Tuned model excelled in accurately predicting positive instances while capturing a higher proportion of actual positives. Furthermore, the AUC-PR of the Fine-Tuned model was 0.80, exceeding the Non-Fine-Tuned AUC-PR curve by 0.08(0.72). This suggested that the Fine-Tuned model exhibited superior overall performance in terms of precision and recall.

d) Random Forest Classifier: An ensemble learning method that constructs numerous decision trees during training and outputs the most frequently occurring classes for classification problems. It enhances the accuracy of sentiment classification tasks by aggregating the results of multiple trees.

On using the embeddings generated by the Non-Fine-Tuned fastText word Embedding model the accuracy obtained is 0.82, whereas the accuracy obtained by using the Fine-Tuned fastText model is 0.87. Resulting in a better accuracy of the Fine-Tuned Model.

The PR curve for the Fine-Tuned model was positioned closer to the top right, signifying its superior classification

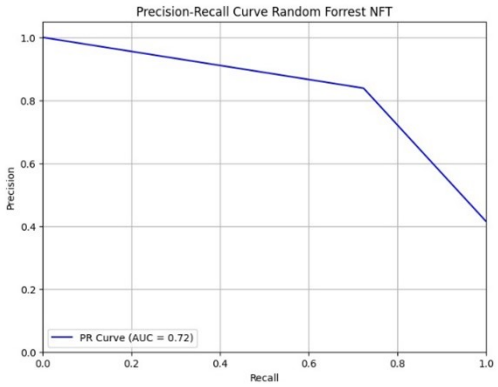

Fig. 8. Random Forest PR Curve for Non-Fine-Tuned Model.

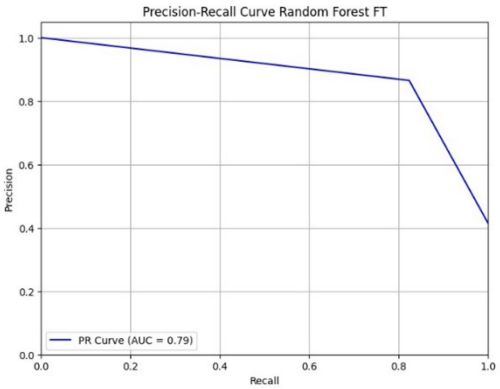

Fig. 9. Random Forest PR Curve for Fine-Tuned Model.

ability in comparison to the Non-Fine-Tuned PR curve. This indicated that the Fine-Tuned model excelled in accurately predicting positive instances while capturing a greater proportion of actual positives. Moreover, the AUC-PR for the Fine-Tuned model, measured at 0.79, outperformed the Non-Fine-Tuned AUC-PR curve by 0.07 (0.72). This underscored the superior overall performance of the Fine-Tuned model in precision and recall.

e) Support Vector Machine: It is a supervised machine learning algorithm. It is used to perform regression as well as classification tasks.It can handle both linear and non-linear relationships in data and is widely used for its versatility and robust performance. It finds an optimal hyperplane in a high-dimensional space that best distinguishes data points of different classes. On using the embeddings generated by the Non-Fine-Tuned fastText word Embedding model the accuracy obtained was 0.79, whereas the accuracy obtained by using the Fine-Tuned fastText model is 0.87. Resulting in a better accuracy of the Fine-Tuned Model.

The Fine-Tuned PR curve, situated closer to the top right, denoted the superior classification prowess of the model compared to the Non-Fine-Tuned PR curve. This suggested that the Fine-Tuned model excelled in precision by accurately predicting positive instances while capturing a larger proportion of actual positives. Additionally, the AUC-PR for the Fine-Tuned

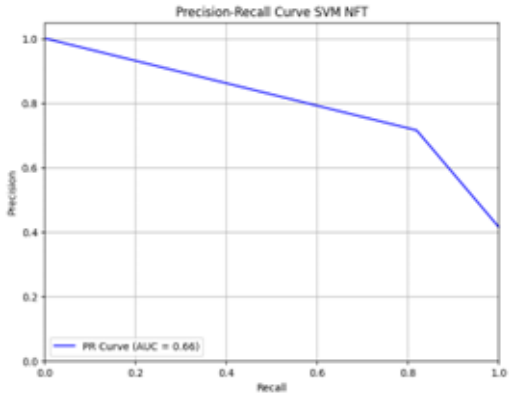

Fig. 10. SVM PR Curve for Non-Fine-Tuned Model.

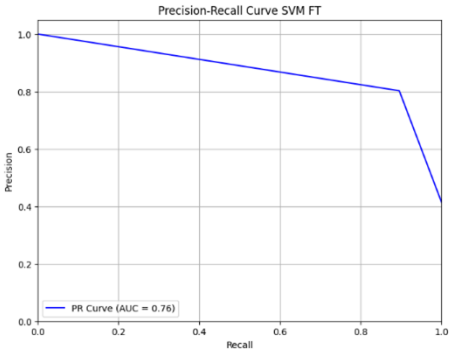

Fig. 11. SVM PR Curve for Fine-Tuned Model.

model achieved a value of 0.76, surpassing that of the Non-Fine-Tuned by 0.10 (0.66). This emphasized the Fine-Tuned model's superior overall performance in terms of precision and recall.

The comparison of the Machine Learning Models made it evident from the elevation of the accuracy of the Fine-Tuned Model that the Fine-Tuned model outperforms the Non-Fine-Tuned model. The highest accuracy of 0.89 was resulted by the K Nearest Neighbours model making it most suitable for classification. Logistic Regression showed a maximum difference of 0.9 in the accuracies of the Non-Fine-Tuned and Fine Tuned Model. Refer to Table I for the comparison of models.

TABLE I
COMPARISON OF ACCURACY

Models	Accuracy	
	Non-Fine-Tuned Model	*Fine-Tuned Model*
Logistic Regression	0.78	0.87
Decision Tree Classifier	0.71	0.76
K-Nearest Neighbors	0.83	0.89
Random Forest Classifier	0.82	0.87
Support Vector Machine	0.79	0.87

Epoch Loss Curve : An epoch loss curve is a graphical representation of the validation loss and the training of a machine-learning model over different epochs.

f) Convolutional Neural Network(CNN): A deep learning algorithm used for processing structured grid data, such as images. It employs convolutional layers to automatically learn hierarchical features and has been highly successful in image recognition tasks. It automatically learns hierarchical features from textual data for sentiment analysis in natural language processing.

The model after getting trained on 20 epochs resulted in an accuracy of 0.8443. The Epoch Loss graph depicted a

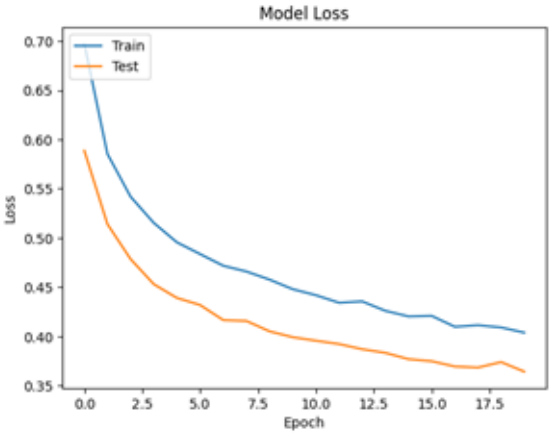

Fig. 12. CNN Epoch-Loss Curve for Fine-Tuned Model.

decline in the loss values as the number of epochs increased, suggesting a considerable reduction in the error between the predicted outputs and the actual targets. The parallel and decreasing trend with low spikes and fluctuations suggested that the model could capture the patterns in the data. The low fluctuations indicated stability in the learning process. The fact that the curves were parallel suggested there was a low variance between the training and validation datasets. The model did not overfit the training data, and it generalized well to new, unseen examples.

g) Recurrent Neural Network(RNN): A type of neural network designed for sequence data, where information from previous time steps is fed into the model. It is able to capture temporal dependencies therefore suitable for NLP and speech Recognition. It captures temporal dependencies in sequential data, enhancing sentiment analysis for text with context.

The model was trained on 10 epochs and post 10 epochs the accuracy obtained was 0.8014.

The Epoch Loss Graph showed a decline in loss values with a growth in the variety of epochs for both the Training and Validation Curves. The steady decrease in the Training curve indicated that the model turned into gaining knowledge effectively from the provided data. Furthermore, the general descending fashion within the Validation curve advised good generalization. Upon looking at the converging nature of each of the Training and Validation loss curves, it was inferred that the training of the model was halted. Slight fluctuations in the graph had been observed, attributed to obstacles in dataset technology and the ability of the embedding model used in the course of training.

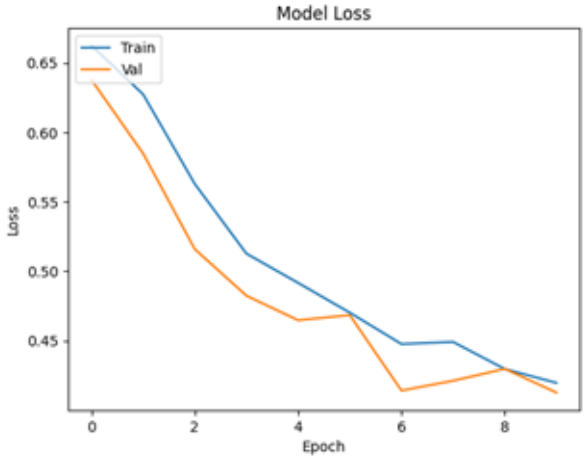

Fig. 13. RNN Epoch-Loss Curve for Fine-Tuned Model.

h) Long Short-Term Memory(LSTM): A specialized kind of RNN that addresses the vanishing gradient trouble, thus helps get information about long-time period dependencies. It's broadly utilized in responsibilities that involve sequential statistics and require reminiscence of beyond occasions. It addresses vanishing gradient hassle, taking into account powerful sentiment evaluation on sequences with long-time period dependencies.

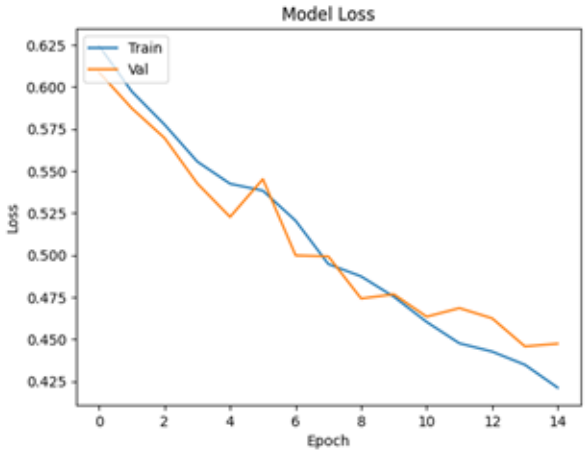

Fig. 14. Bi-LSTM Epoch-Loss Curve for Fine-Tuned Model.

The Epoch-Loss Graph illustrated a decreasing trend in both the training and validation curves, signifying a decline in loss as the model progressed through training. This pointed to successful learning from the training data and the model's capability to generalize to the validation set. The close alignment of training and validation losses suggested the model's effectiveness in generalizing to new, unseen data, indicating a positive avoidance of overfitting to the training set. Bi-LSTM outperforms LSTM as comparatively smooth curves were observed in the Bi-LSTM graph also the training and validation curves were close to convergence which results in a better performance of the model.

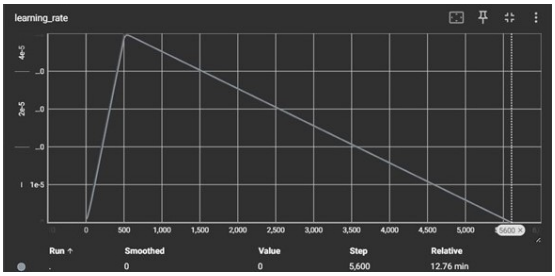

Fig. 15. RoBERTa Learning Rate Curve for Fine-tuned Model.

i) Robustly optimized BERT approach(RoBERTa): A transformer-based natural language processing model that is an extension of BERT (Bidirectional Encoder Representations from Transformers).It is capable of applying NLP techniques on huge datasets and then fine-tuning them for specific applications. It could also captured nuanced language representations.

Table II showcases the confusion matrix of the pre-trained Roberta model when evaluated on the test dataset. The number of tuples misclassified is considerable in the count, indicating the model is not able to handle the slang well.

TABLE II
CONFUSION MATRIX FOR PRETRAINED ROBERTA MODEL

	Predicted Positive	**Predicted Negative**
Actual Positive	1157	244
Actual Negative	395	855

The learning rate started from a minimum value(0) and linearly increased during the first part of the cycle as it reached 500 cycles. It then linearly decreased during the next part of the cycle. This process was repeated for a fixed number of cycles. Therefore it depicted a triangular learning rate policy, it helped the model explore a larger part of the loss landscape by initially using a low learning rate, then gradually increasing it to cover a wider range, and finally decreasing it to converge to a good minimum.

The initial phase of the Loss Curve showed an exponential decrease therefore indicating a rapid learning and adaptation of the model to the training data. The loss steadily decreased, therefore showing improvement. The plateaus in between and in the end conveyed that the model has converged to a local minimum, but the further decrease in the loss suggested the improved performance of the model.

Table III showcases the confusion matrix of the Roberta model fine-tuned on the training set and then evaluated on the test dataset. The matrix shows significant improvement over the pre-trained model with a very low count of misclassified tuples.

TABLE III
CONFUSION MATRIX FOR ROBERTA FINE-TUNED MODEL

	Predicted Positive	**Predicted Negative**
Actual Positive	1467	31
Actual Negative	85	1068

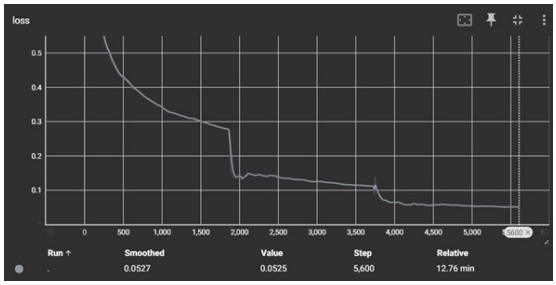

Fig. 16. RoBERTa Loss Curve for Fine-tuned Model.

j) Comparison of models: A few examples depicting the comparison of the outputs from the pre-trained - twitter-roberta-base-sentiment model[13] and authors' Fine-Tuned-Model - roberta-finetune-slangs[12] are stated in Table IV. As per the results, the first sentence used the slag ftw- "for the win" which is a Positive sentence but it was misclassified as neutral by the pre-trained twitter-roberta-base-sentiment model[13]. The second sentence "I h8 that person" can also be written without the slang as "I hate that person" which is a negative sentence. It was classified correctly by both the models as negative but fine-tuned model classifies it as negative with a higher sentiment score than the pre-trained model.

TABLE IV
COMPARISON OF MODELS

Sentence	twitter-roberta-base-sentiment			roberta-finetune-slangs	
	Positive	*Negative*	*Neutral*	*Positive*	*Negative*
Team India ftw	0.109	0.039	0.852	0.928	0.072
I h8 that person	0.093	0.599	0.308	0.193	0.806

V. CONCLUSION

This study highlights the critical role of incorporating net slang and abbreviations into NLP programs to enhance their understanding of informal digital conversations. With a slang dictionary created using selective slang, short phrases, and FastText embedding strategies, exploratory sentiment analysis, the approach demonstrated remarkable improvements in accuracy. It provides empirical evidence of a transformative ability to hold coincident linguistic features to successfully navigate the complexity of digital conversations. The integration of slang enriches NLP's contextual information and leads to a more nuanced analysis of users' emotions in digital contexts.

VI. FUTURE SCOPE

The research completed till now specifically concentrates on advancing the conventional Natural Language Processing(NLP) Tasks, incorporating informal expressions. The utilization of slang and short words is prevailing in digital communication, opening greater possibilities for further exploration and research. This research can be expanded further with the use of recent and cutting-edge slang and curating a larger dataset with more sentences for each informal expression to increase accuracy. Due to the restrictions of the platform, FastText word embedding proved to be appropriate which furnished top-notch outcomes, while other advanced embedding models can also be deployed onto the dataset to optimize the accuracy. The usage of slang and abbreviations is escalating and is anticipated to increase in the coming days, highlighting the need for incorporating it within the NLP tasks and broadening the research area.

REFERENCES

[1] Tan, Kian Long, Chin Poo Lee, and Kian Ming Lim. "A survey of sentiment analysis: Approaches, datasets, and future research." Applied Sciences 13.7 (2023): 4550.https://doi.org/10.3390/app13074550.

[2] Kiritchenko, Svetlana, Xiaodan Zhu, and Saif M. Mohammad. "Sentiment analysis of short informal texts." Journal of Artificial Intelligence Research 50 (2014): 723-762.

[3] Mishra, Jayant. (2023). "TWITTER SENTIMENT ANALYSIS." INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 07.10.55041/IJSREM24071.

[4] Farogh, Sakhawia. (2023). A Comprehensive Evaluation and Comparative Analysis of Data Mining Techniques for Sentiment Analysis in Social Media. International Journal of Advanced Research in Science, Communication and Technology. 40-45. 10.48175/IJARSCT-11609.

[5] H. P. Patil and M. Atique, "Sentiment Analysis for Social Media: A Survey," 2015 2nd International Conference on Information Science and Security (ICISS), Seoul, Korea (South), 2015, pp. 1-4, doi: 10.1109/ICISSEC.2015.7371033. keywords: Sentiment analysis;Media;Twitter;Context;Classification algorithms;Data mining;Analytical models,

[6] Dey, Paramita and Dey, Soumya (2023) "SENTIMENT ANALYSIS OF TEXT AND EMOJI DATA FOR TWITTER NETWORK," Al-Bahir Journal for Engineering and Pure Sciences: Vol. 3: Iss. 1, Article 1.

[7] Sharma, V., Srivastava, S., Valarmathi, B., Srinivasa Gupta, N. (2021). A Comparative Study on the Performance of Deep Learning Algorithms for Detecting the Sentiments Expressed in Modern Slangs. In: Bindhu, V., Tavares, J.M.R.S., Boulogeorgos, AA.A., Vuppalapati, C. (eds) International Conference on Communication, Computing and Electronics Systems. Lecture Notes in Electrical Engineering, vol 733. Springer, Singapore.

[8] Gupta, S., Bisht, S., Gupta, S. (2021). Sentiment Analysis of an Online Sentiment with Text and Slang Using Lexicon Approach. In: Satapathy, S.C., Bhateja, V., Favorskaya, M.N., Adilakshmi, T. (eds) Smart Computing Techniques and Applications. Smart Innovation, Systems and Technologies, vol 224. Springer, Singapore. https://doi.org/10.1007/978-981-16-1502-3_11.

[9] K. N. Prasanthi, R. Eswari Madhavi, D. N. Sai Sabarinadh and B. Sravani, "A Novel Approach for Sentiment Analysis on social media using BERT & ROBERTA Transformer-Based Models," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-6, doi: 10.1109/I2CT57861.2023.10126206.

[10] Manuel, K., Kishore Varma Indukuri, and P. Radha Krishna. "Analyzing internet slang for sentiment mining." 2010 second Vaagdevi international conference on information Technology for Real World Problems. IEEE, 2010.

[11] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in Pre-Training Distributed Word Representations. ArXiv. /abs/1712.09405.

[12] roberta-finetune-slangs: Hugginface, Jan. 2023. [Online]. Available: https://huggingface.co/spectre0108/roberta-finetune-slangs.

[13] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1644–1650, Online. Association for Computational Linguistics.