DATA255 | Team 6 Project Report | Cyberbullying Text Classification Using Deep Learning Models

Dharani Aningi
Dept. Applied Data Science
SJSU
San Jose, CA
Email: dharani.aningi@sjsu.edu

Gyana Apuroopa Chilakalapudi
Dept. Applied Data Science
SJSU
San Jose, CA
Email: gyanaapuroopa.chilakalapudi@sjsu.edu

Sai Keerthana Kattige
Dept. Applied Data Science
SJSU
San Jose, CA
Email: saikeerthana.kattige@sjsu.edu

***Abstract*—Address the increasing surge in cyberbullying space, our project leverages advanced deep learning models to accurately detect and categorize the online hate speech, aiming to provide a safer environment for everyone to express their thoughts**

I. Motivation

In the modern era where individuals have the ability and tools to share and express their thoughts freely over the internet, there is a very high likelihood that the attackers and bullies would attack and comment on an individual's speech. This bullying and attack on freedom of speech reduces the confidence of an individual expressing their thoughts freely on public forums. There are very minimal tools available on the internet to safeguard hate speech and bullying. Our project aims to address this problem by classifying the speech and comments of an individual in real time and flag improper text. This project aims to classify the text posted on the internet as positive, negative, neutral or irrelevant.

In this project we leveraged the latest advancement in deep learning as tools to classify the text. This also provides a proactive response to protect individuals from online harassment and protect their Right to Speech. We also aim to create a safer online space for individuals where they can freely express their thoughts and protect their mental well being.

We were also victims of cyberbullying at some point of time in our life and as a team we aim to tackle and solve this problem by contributing back to the internet and making it a safe space. This endeavor will help individuals to protect their freedom of expression.

II. Background

In this growing digital era where individuals crave for connectivity, cyberbullying has become more prevalent. Recent studies conducted by Cyberbullying Research Center highlights the importance of this problem. This study revealed that more than 37% of young generations are affected by cyberbullying and over 30% of these individuals are prone to repeated bullying. Another research conducted by UNICEF also emphasizes the importance of this problem by indicating that 1/3rd of the world's young population have been affected by this cyberbullying. The Internet has provided a medium for bullies to harass the younger generation. COVID-19 pandemic have taken this problem to the next level, where the young generation relied on the internet heavily to express their thoughts and any social interactions. This surge in usage of the internet also provided an opportunity for cyberbullies to exploit vulnerabilities increasing a negative impact on individuals. Individuals and cyberbullies spent a lot of time on the internet for social connection and used the social bonding and internet as a medium to attack individuals. Our project aims to address this problem by leveraging real time internet data to train the models and classify the hate speech.

III. Literature Survey

TABLE I
Relevant Publications

Paper	Models Used	Results
Cyberbullying Detection Using Deep Neural Network - 2021 (Md Faisal Ahmed)	RandomForest, SVM, KNN, Naive Bayes	SVM algorithm stood out with an improved accuracy of 85%
Cyberbullying Detection in Social Networks - Reproducibility Study	DNN, CNN, LSTM, BLSTM	This study shows that the DNN models were adaptable and transferable to the new dataset.
Detecting Cyberbullying with Text Classification	1DCNN with GloVe embeddings, SVM, Logistic Regression, LSTM, BERT, RNN	1DCNN with GloVe word embeddings performed better for cyberbullying detection
Classification of Covid-19 tweets using deep learning techniques	CNN, RNN, RCNN, RNN+LSTM, BI-LSTM+Attention, GloVe, Word2Vec	RNN with Bidirectional LSTM model achieved an accuracy of 93%

- We are doing a multi classification problem in classifying the given text real time and flag the content with target variables. We also considered keeping multiple classes as this would help us in preventing over-fitting or under-fitting the model.
- The aim of our model is to prevent cyberbully by removing inappropriate content from the comments section.

IV. Methodology

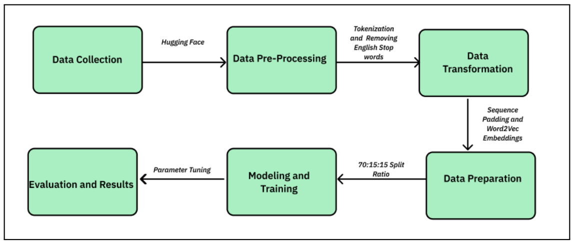


Fig. 1. Project Work Flow

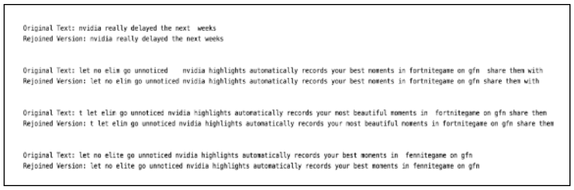
Figure 1 outlines a data mining process from 'Data Collection' to 'Evaluation,' ensuring a systematic flow through preprocessing, EDA, and modeling for insightful data analysis.

A. Data Collection

The dataset for sentiment analysis is sourced from Hugging Face, consists of 74,315 entries with 69,156 unique texts, offering a rich variety for analysis. It includes four distinct sentiment categories across 70,210 unique prompts, making it highly suitable for training sophisticated deep learning models.

B. Data Preprocessing

In the data preprocessing phase for our sentiment analysis project, we first addressed 681 null instances in the 'text' column, converting all texts to lowercase for uniformity.Next by removing numbers and special characters,which do not contribute to sentiment. Then tokenizing the text into words, and excluding common English stop words. Then applied stemming to simplify the text to its root forms.These processed tokens are rejoined to reconstruct the text for further analysis.

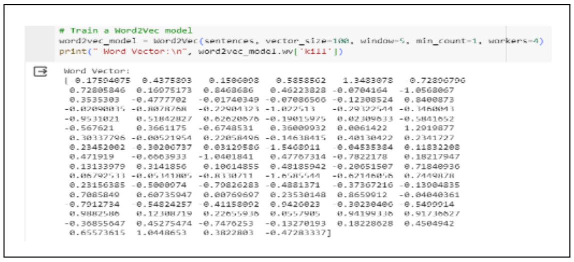


```
Original Text: nvidia really delayed the next  weeks
Rejoined Version: nvidia really delayed the next weeks


Original Text: let no elim go unnoticed    nvidia highlights automatically records your best moments in fortnitegame on gfn  share them with
Rejoined Version: let no elim go unnoticed nvidia highlights automatically records your best moments in fortnitegame on gfn share them with


Original Text: t let elim go unnoticed nvidia highlights automatically records your most beautiful moments in  fortnitegame on gfn share them
Rejoined Version: t let elim go unnoticed nvidia highlights automatically records your most beautiful moments in fortnitegame on gfn share them


Original Text: let no elite go unnoticed nvidia highlights automatically records your best moments in  fennitegame on gfn
Rejoined Version: let no elite go unnoticed nvidia highlights automatically records your best moments in fennitegame on gfn
```

Fig. 2. After Rejoining Tokens

C. Data Transformation

In the data transformation phase for deep learning models, the primary objectives are to establish standardized input dimensions,convert textual and categorical data into suitable numerical formats.Deep Learning models require consistent input dimensions.Also, unequal sequence lengths can lead to data loss or computational inefficiency.To standardize input dimensions, we set the sequence length to the 75th percentile of text lengths, plus a buffer, ensuring uniformity across texts. We employed Word2Vec for tokenization and generating word embeddings to capture word semantics and reduce dimensionality.It captures the context in which words appear in sentences or documents and assigns similar numerical vectors to words that tend to appear in similar contexts exemplifying transforming words like "kill" into embeddings.For model compatibility, we created an embedding matrix,this matrix serves as a lookup table where each row representing a Word2Vec vector and used label encoding to convert categorical features into numerical formats.



```
# Train a Word2Vec model
word2vec_model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4)
print(" Word Vector:\n", word2vec_model.wv['kill'])
```

```
Word Vector:
[-0.17594075  0.4375803   0.1506008   0.5858562   1.3483078  -0.72896796
  0.72805846  0.16075173  0.8468686   0.46223828 -0.0704164  -1.0568067
  0.3535303  -0.4777702  -0.01740349 -0.07086566 -0.12308524  0.8400873
 -0.02090035 -0.3078768  -0.22904323 -1.022513   -0.29322544 -0.3460043
 -0.9531021   0.51842827  0.62620676 -0.19015975  0.02309633 -0.5841652
 -0.567621    0.3661175  -0.6748531   0.36000932  0.0061422   1.2019877
  0.30337796 -0.00521054  0.22058496 -0.14638415  0.40130422  0.2341727
  0.23452002 -0.30206737  0.03120586 -1.5468011  -0.04535384  0.11832208
  0.471919   -0.0663933  -1.0401841   0.47767314 -0.7822178   0.18217947
  0.13133979  0.3141856   0.10614855  0.48185942 -0.20651507  0.71840936
  0.06792533 -0.05341005 -0.8330711  -1.6505544  -0.62146056  0.7449078
  0.23156385 -0.5000074  -0.70826283 -0.4881371  -0.37367216 -0.13004835
  0.7085849   0.60735947  0.00769697  0.23530148  0.8659912  -0.04040361
 -0.7912734  -0.54824257 -0.41158092  0.9420023  -0.30230406 -0.5499914
  0.9882586   0.12308719  0.22655936  0.0557905   0.94199336  0.91736627
 -0.36855647  0.45275474 -0.7476253  -0.13270193  0.18228628  0.4504942
  0.65573615  1.0448653   0.3822803  -0.47283337]
```

Fig. 3. Utilizing Word2Vec for Tokenization and Word Embeddings Creation

V. Modeling and Training

A. Recurrent Neural Network (RNN)

The RNN model includes an embedding layer, an RNN layer, a fully connected layer, and a dropout layer. The training is performed using the AdamW optimizer with a learning rate of 1e-3, 30 epochs and a batch size of 64. The soft-max function is applied for cross-entropy loss.The early stopping criterion checks if the validation loss increases consecutively, at which point training is halted, the training topped at epoch 9 with the accuracy of 30.36. The below fig 5 show the training validation accuracy's and losses, we see that the accuracy is too low.

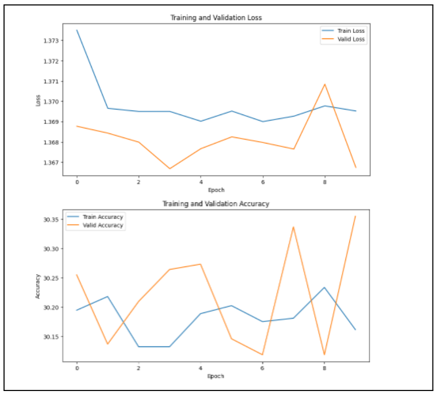


Fig. 4. Training and Validation Accuracy and Loss for RNN

B. Long Short Term Memory(LSTM)

The LSTM model, comprises of an embedding and LSTM layers for sequential analysis, a fully connected layer, and dropout for regularization, is configured with specific hyper parameters. Training involves 30 epochs, where the model's performance stabilizes with a training loss of approximately 1.370 and an accuracy of around 30.26%. Validation metrics consistently report a loss of 1.366 and an accuracy of 30.32%. However, at Epoch 19, early stopping is triggered due to increasing validation loss, resulting in a final training loss of 1.359 with a training accuracy of 30.80% and a final validation loss of 1.366 with a validation accuracy of 30.47%. Despite multiple epochs, the model's performance remains limited, highlighting the need for further enhancements.The training and validation curves illustrate that the model's performance plateaus after a certain point, and the early stopping mechanism effectively prevents overfitting.

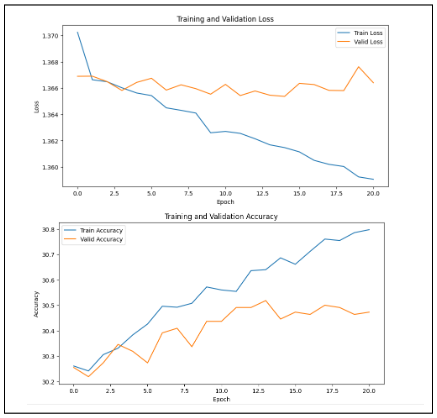


Fig. 5. Training and Validation Accuracy and Loss for LSTM

C. *Bi-Directional Gated recurrent units (GRU)*

Bi-directional GRU model included an embedding layer, a bi-directional GRU layer for context analysis, and a fully connected layer is used for classification. To combat overfitting, we used dropout regularization and the AdamW optimizer for efficient loss minimization,incorporated early stopping. With an initial dropout rate of 0.3, the model exhibited overfitting, marked by a high training accuracy of 88% against significantly lower validation and test accuracy's (60.2% and 59%, respectively). Upon increasing the dropout rate to 0.5, we observed an improvement in generalization, evidenced by a closer alignment of training (77.53%) and validation (62.46%) accuracy's, indicating a more robust model performance.

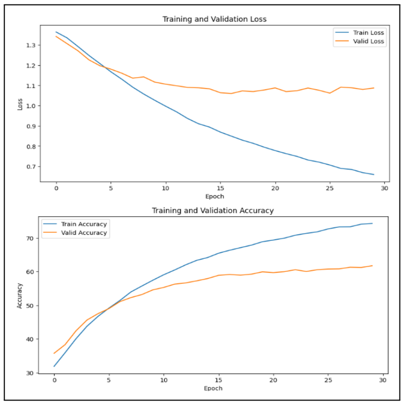


Fig. 6. Training vs Validation Accuracy and Loss with dropout = 0.5 and no early stopping (Bi-GRU)

D. *BERT*

In order to classify the text into multiple classifications, we leveraged BERT(Bidirectional Encoder Representations from Transformers) architecture.This model consists of a transformer which captures the self attention and feed forward neural network which would capture and correct the context of the given text.This model is already pre-trained on a huge amount of text which helps us in classifying the text. The above figure shows the visualization of how BERT processes and prioritizes different elements of the input sequence-"Come meet one of the lovely Gaming Goddesses" , helps in understanding which words or sub words significantly contribute to comprehending the input context and semantics.The model tokenizer and allows the model to yield attention values, representing the weightings assigned to various segments of the input and reveal the token relationships and indicating the focus of each token during encoding. Two input sentences, "Come meet one of the lovely Gaming Goddesses" and "All the Borderlands are damn rubbish," are given to the model which are tokenized, encoded, and fed into a BERT model for attention visualization. The above figure shows an interactive heat map, offering insights into the attention patterns of the BERT model. This enables a deeper understanding of how the model processes and prioritizes different aspects of the input text, aiding in the interpretation of its attention mechanisms.

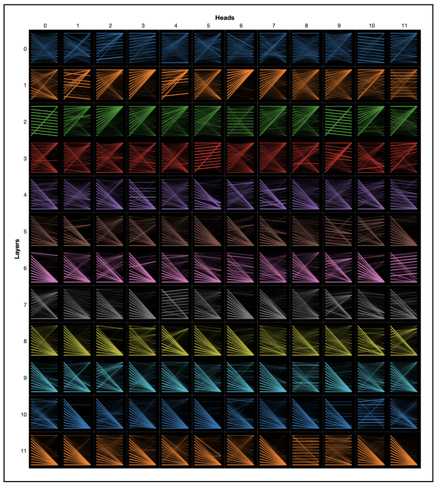


Fig. 7. Visualization of Weights of an input Sequence of BERT model

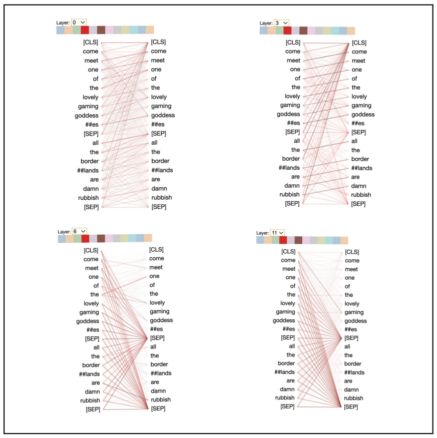


Fig. 8. Visualization of Weights interaction of Bert model

VI. Evaluation and Results

A. Bidirectional GRU

The incorporation of early stopping in our model was a strategic decision to address overfitting. This technique halts training when the validation loss fails to show improvement throughout five epochs. As a result of implementing early stopping, we noticed a moderate decline in training accuracy to 71.80%, while the validation and test accuracy's recorded were 61.07% and 60.34% respectively. This change indicated a more balanced approach to the bias-variance trade-off, suggesting that the model was not over-learning the training data.

To enhance the model's ability to generalize further, we increased the dropout rate to 0.7, maintaining the early stopping criterion. This adjustment led to a significant shift in the model's performance: the training accuracy decreased to 58.82%, whereas the validation and test accuracy's were observed at 55.48% and 54.38%, respectively. The considerable decrease in training accuracy, coupled with the relatively stable validation and test accuracy's, pointed towards a stronger resistance to overfitting. This was a clear indication that higher levels of regularization, achieved through increased dropout, effectively minimized overfitting. However, this also raised concerns about the model potentially underfitting the training data, as evidenced by the substantial drop in training accuracy.

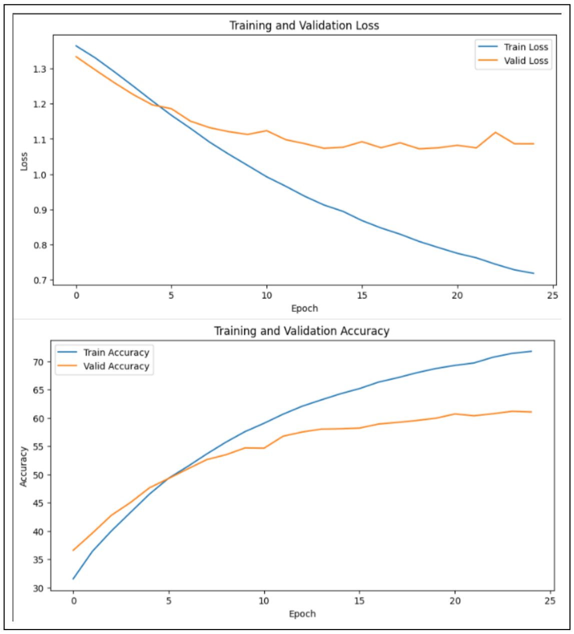


Fig. 9. Training vs Validation loss and accuracy with dropout = 0.5 and early stopping (Bi-GRU)

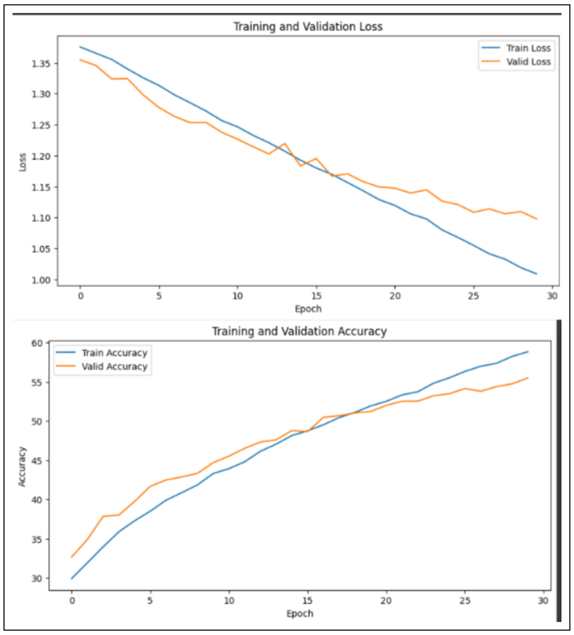


Fig. 10. Training vs Validation loss and accuracy with dropout = 0.7 and early stopping (Bi-GRU)

B. LSTM

We explored the effectiveness of a Bidirectional LSTM (Bi-LSTM) variant as an alternative to the standard LSTM architecture.The training procedure involves utilizing a Bi-LSTM model with a dropout rate of 0.5 for regularization. The loss function employed is CrossEntropyLoss, and optimization is achieved using the AdamW optimizer with a learning rate of 1e-3. The results reveal that as training progresses, both training and validation metrics improve. The initial training loss is 1.356 with an accuracy of 32.43%, while the validation loss is 1.336 with an accuracy of 35.99%. However, by Epoch 23/30, early stopping is invoked due to an increasing validation loss, indicating no further improvement. The final training loss is 0.725 with an accuracy of 71.83%, while the final validation loss is 1.148 with an accuracy of 60.37%.

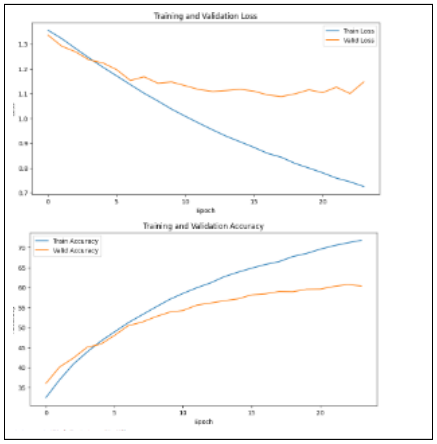


Fig. 11. Training and Validation Accuracy and Loss of Bi-LSTM

C. BERT

As part of fine tuning this model for our specific project we trained the bert model with 10 epochs and a batch size of 32 with a learning rate of 1e-5. During training Bert model utilizes the attention masks to identify and consider the tokens for each iteration. To preserve uniformity across the tokens we also implemented padding to normalize the tokens. Since this model uses attention masks this would result in high accuracy and high training time. The training data is sent over to this model using AdamW optimization technique and the model parameters are calculated internally using back propagation mechanisms.We then validated the trained model using a validation dataset which resulted in an accuracy of 96% with training and validation losses 0.399 and 0.302 respectively.The fig 6 and 7 show the training and validation accuracy's and losses.

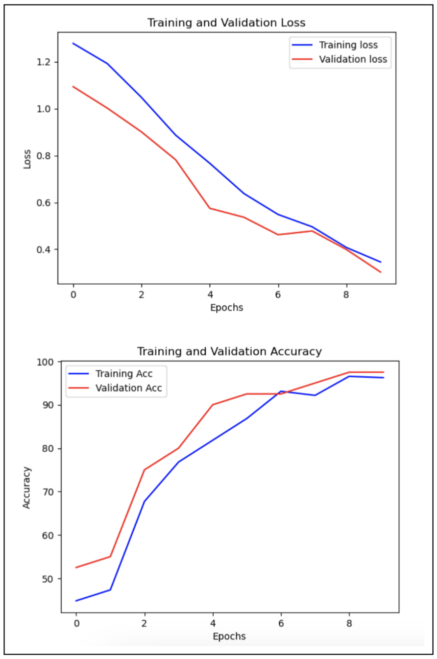
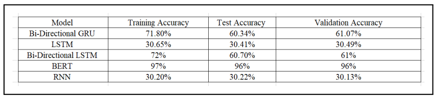


Fig. 12. Training and Validation Accuracy's of Bert

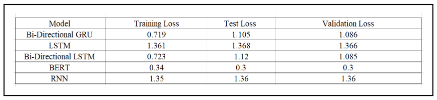
VII. Conclusion

In this project, BERT emerged as the superior model, delivering impressive train and validation accuracies of 97% and 96%, respectively, highlighting its advanced capability for nuanced language understanding. The Bi-Directional GRU and LSTM models performed on par with each other, with both achieving 71% training accuracy and 61% validation accuracy, reflecting their competence but also suggesting potential areas for optimization. Conversely, the RNN model proved inadequate, failing to produce scalable or reliable results for the complexity of hate speech detection. These outcomes underscore the necessity of employing more sophisticated models like BERT for tasks requiring deep linguistic analysis and contextual understanding to effectively distinguish hate speech.



Model	Training Accuracy	Test Accuracy	Validation Accuracy
Bi-Directional GRU	71.80%	60.34%	61.07%
LSTM	30.65%	30.41%	30.49%
Bi-Directional LSTM	72%	60.70%	61%
BERT	97%	96%	96%
RNN	30.20%	30.22%	30.13%

Fig. 13. Model Accuracies



Model	Training Loss	Test Loss	Validation Loss
Bi-Directional GRU	0.719	1.105	1.086
LSTM	1.361	1.368	1.366
Bi-Directional LSTM	0.723	1.12	1.085
BERT	0.34	0.3	0.3
RNN	1.35	1.36	1.36

Fig. 14. Model Losses

VIII. Discussion and Future Scope

A. Integration with Real-Time Platforms

The integration of cyberbullying detection models into social media and online forums in real-time marks a significant advance in preventing cyberbullying. This approach enables immediate identification and mitigation of harmful content. Future developments could focus on creating partnerships with major platforms and developing APIs for seamless integration, with a strong emphasis on data privacy and ethical usage.

B. Early Detection and Emotional Analysis

Enhancing cyberbullying detection systems to identify subtle and early signs of negative behavior through advanced emotional and contextual analysis is crucial for prompt intervention and fostering empathetic online communities. Future efforts could focus on integrating sophisticated sentiment analysis and algorithms capable of deciphering nuances in human communication, such as sarcasm and indirect aggression.

C. Multilingual and Cultural Adaptation

Cyberbullying often involves language that is deeply rooted in cultural contexts.What is considered a harmless phrase in one culture may be deemed a terrible insult in another.Therefore,models must be culturally sensitive, understanding and interpreting language within the context of its cultural context. Each language has its own mix of idioms, slurs, and colloquialisms, which can make training a model on data from another language difficult.The ability to accurately detect harmful content is heavily dependent on understanding these nuances. Dialectical variations can cause major variances in meaning and interpretation even within the same language. Recognizing these variances is critical for detecting cyberbullying accurately. Using cutting-edge NLP techniques with a good grasp of semantic comprehension can significantly improve the model's capacity to process and understand material in different languages.Transformer-based models (e.g., BERT, GPT) can be fine-tuned for specific languages and dialects after being pre-trained on huge,diverse corpus.Building a comprehensive dataset that includes a wide range of languages, dialects, and cultural contexts.

D. Transfer Learning

Transfer learning improves cyberbullying detection by fine-tuning pre-trained language models such as BERT or GPT on specific datasets.This process adapts their powerful semantic understanding of text to the nuances of identifying cyberbullying, requiring less labeled data than training from scratch.Fine-tuning also enables hyperparameter adjustment to improve model performance, with careful evaluation required. It's an iterative process that can lead to a very effective and adaptive cyberbullying detection system that can handle a wide range of writing styles, languages, and internet platforms.

REFERENCES

[1] Ahmed, M. F., Mahmud, Z., Biash, Z. T., Ryen, A. a. N., Hossain, A., Ashraf, F. B. (2021). Cyberbullying Detection Using Deep Neural Network from Social Media Comments in Bangla Language. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2106.04506

[2] Dadvar, M., Eckert, K. (2018). Cyberbullying detection in social networks using deep learning based models; A reproducibility study. arXiv (Cornell University). https://arxiv.org/pdf/1812.08046.pdf

[3] Sangeethapriya, R., Akilandeswari, J. (2022). Detecting Cyberbullying with Text Classification Using 1DCNN and Glove Embeddings. In Detecting Cyberbullying with Text Classification Using 1DCNN and Glove Embeddings (pp. 179–195). https://doi.org/10.1007/978-981-19-3015-7.14

[4] Sunagar, P., Kanavalli, A., Poornima, V., Hemanth, V. M., Sreeram, K., Shivakumar, K. (2021). Classification of Covid-19 tweets using deep learning techniques. In Lecture notes in networks and systems (pp. 123–136). https://doi.org/10.1007/978-981-16-1395-1.10

[5] Al-Garadi, M. A., Hussain, M. R., Khan, N., Murtaza, G., Nweke, H. F., Ali, I., Mujtaba, G., Chiroma, H., Khattak, H. A., Gani, A. (2019). Predicting cyberbullying on social media in the big data era using Machine learning Algorithms: Review of literature and open challenges. IEEE Access, 7, 70701–70718. https://doi.org/10.1109/access.2019.2918354

[6] Yuvaraj, N., Srihari, K., Dhiman, G., Somasundaram, K., Sharma, A., Rajeskannan, S., Soni, M., Gaba, G. S., AlZain, M. A., Masud, M. (2021). Nature-Inspired-Based approach for automated cyberbullying classification on multimedia social networking. Mathematical Problems in Engineering, 2021, 1–12. https://doi.org/10.1155/2021/6644652

[7] Murshed, B. a. H., Suresha, Abawajy, J., Saif, M. a. N., Abdulwahab, H. M., Ghanem, F. A. (2023). FAEO-ECNN: cyberbullying detection in social media platforms using topic modelling and deep learning. Multimedia Tools and Applications, 82(30), 46611–46650. https://doi.org/10.1007/s11042-023-15372-3

[8] Elsafoury, F., Katsigiannis, S., Pervez, Z., Ramzan, N. (2021). When the timeline meets the pipeline: A survey on Automated Cyberbullying Detection. IEEE Access, 9, 103541–103563. https://doi.org/10.1109/access.2021.3098979

[9] Raj, C., Agarwal, A., Bharathy, G., Narayan, B., Prasad, M. (2021). Cyberbullying Detection: hybrid models based on machine learning and natural language processing techniques. Electronics, 10(22), 2810. https://doi.org/10.3390/electronics10222810

[10] Maslej-Krešňáková, V., Sarnovský, M., Butka, P., Machová, K. (2020). Comparison of Deep Learning models and various Text Pre-Processing techniques for the toxic comments classification. Applied Sciences, 10(23), 8631. https://doi.org/10.3390/app10238631

[11] Machová, K., Mach, M., Adamišín, K. (2022). Machine learning and lexicon approach to texts processing in the detection of degrees of toxicity in online discussions. Sensors, 22(17), 6468. https://doi.org/10.3390/s22176468