University of New Haven Tagliatelle college of Engineering

Masters in data science

Course: Natural Language Processing

Title: Cyberbullying Detection using Natural language processing

By

Rahul Muvvala

Ajay Kumar Ganipineni

Kalpana Chamala

Under the guidance of,

Prof. Vahid Behzadan, Ph.D.

vbehzadan@newhaven.edu

Contents

Abstract:

Natural language processing (NLP) can be used to help detect and prevent cyberbullying, which is an issue that is becoming more and more prevalent in today's digital environment. Thce goal of this project is to apply NLP methods to create a model that can precisely identify instances of cyberbullying by using BLSTM architecture and Glove.

Introduction:

Cyberbullying is a prevalent problem on social media that can be quite upsetting and troubling. It can take on many different forms, but most social networks tend to see it as text. Automated detection of these situations requires the use of sophisticated and intelligent technologies. Existing tools for detecting cyberbullying have a lot of constraints. They only touch on one subject, which is cyberbullying. Additionally, they rely on profanity. We demonstrate how deep learning-based models can get beyond these roadblocks.

To effectively train our models, we want to employ publicly accessible datasets, each of which covers a different aspect of cyberbullying. Examples of racism can be seen, for instance, in the Twitter dataset. Additionally, there exist statistics that include instances of sexism. We train our model utilizing Deep Neural Network (DNN) based models like BLSTM. Word embedding is used to model words in order to capture the semantics and similarities between words. Global Vector (GloVe) was selected for word embedding because it has outperformed other models in named entity recognition and word similarity. Users who publish content that promotes cyberbullying will be reported to the helpline through mail. The algorithm also examines posts about cyberbullying that are viewable by the user.

Natural Language Processing:

Natural Language Processing (NLP) has made considerable strides recently and has grown tremendously in practical significance. These days, NLP is essential to many fields, including information extraction, sentiment analysis, chatbots, and virtual assistants. NLP techniques have improved in accuracy and effectiveness as vast volumes of text data have been more widely available and complex deep learning structures like Transformers have been developed. This advancement has revolutionized fields including customer service, healthcare, banking, and marketing by paving the door for better language understanding, automated text generation, and greater human-machine interaction.

Foundation:

For the purpose of processing natural language, our project combined the GloVe (Global Vectors) embedding method with a Bidirectional Long Short-Term Memory (BLSTM) neural network. A LSTM (Long Short-Term Memory) network version with outstanding performance in sequential data analysis is the BLSTM model. In order to train our BLSTM network, which included two hidden layers with a total of 128 units each, we employed the Adam optimization algorithm.

LSTM's

Recurrent neural networks (RNNs) of the type of Long Short-Term Memory (LSTM) are created to address the vanishing gradient problem, which is a prevalent problem in conventional RNNs. The gradient signal shrinks steadily as it backpropagates through time, which is known as the "vanishing gradient problem," making it challenging for models to learn long-term relationships.

To selectively forget or remember data from earlier time steps, LSTM employs a set of gates. The gates are made up of a sigmoid function that outputs a value between 0 and 1, indicating how much information should be remembered or forgotten. The input gate, forget gate, and output gate are some of the gates.

The input gate regulates how much fresh input should be permitted into the network's memory, or cell state. Which data from the previous cell state should be forgotten is decided by the forget gate. Finally, the output gate chooses what data from the cell state should be output as the prediction.

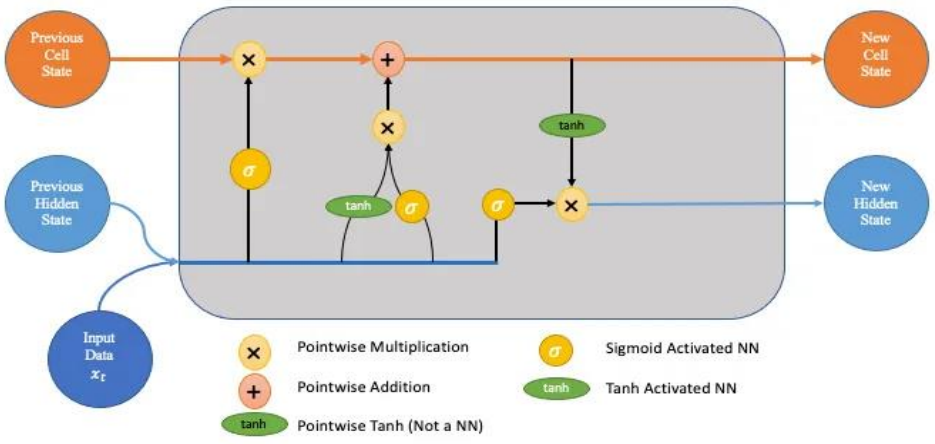


Fig 1: LSTM Architecture

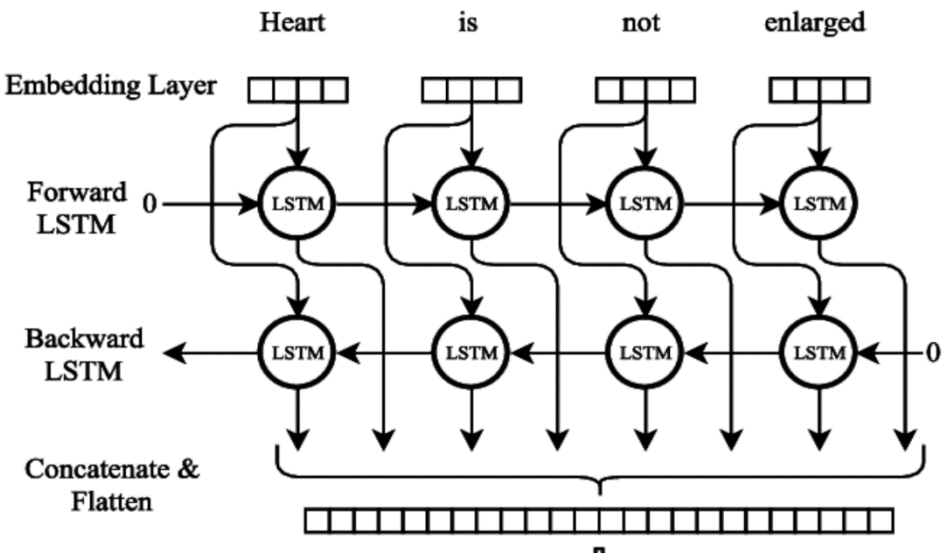


Fig 2: BiLSTM Explained

GloVe for Vectorization:

"GloVe: Global Vectors for Word Representation" Authors: Jeffrey Pennington, Richard Socher, Christopher D. Manning Published: 2014: https://nlp.stanford.edu/pubs/glove.pdf

A pre-trained word embedding method that captures the semantic and syntactic meaning of words was also included in our analysis. GloVe has demonstrated to be very successful in NLP tasks and learns word embeddings using a co-occurrence matrix. In our model, we made use of the 100-dimensional GloVe vectors that were trained on 6 billion words.

In natural language processing (NLP), a word embedding paradigm known as GloVe embeddings, or Global Vectors for Word Representation, is frequently employed. Since each word is represented by a vector of 100 real-valued values, the "100d" in "100d GloVe embeddings" refers to the dimensionality of the embeddings. These embeddings, which capture the semantic and syntactic links between words, are created through training on huge corpora of text data. The 100d GloVe embeddings strike a balance between computing effectiveness and the ability to capture fine-grained semantic nuances. They have been widely used in many NLP tasks, including sentiment analysis, named entity recognition, and text categorization, and they provide important contextual information for better language understanding and downstream tasks.

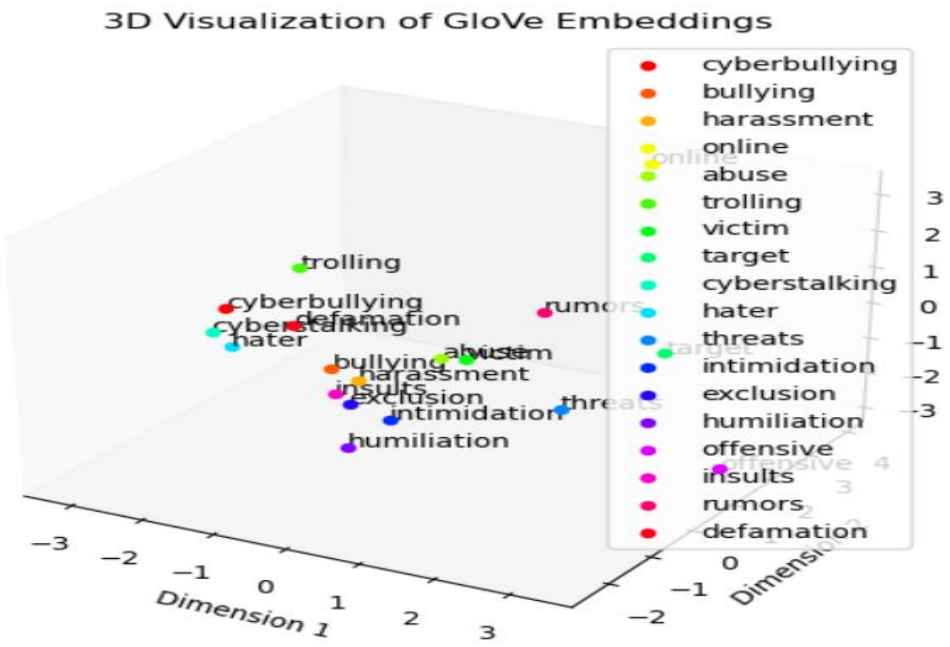


Fig 3: GloVe Embedding Visualization

Methodology:

1. Data Set

The dataset we used for cyberbullying detection contains 16090 Tweets

	id	tweet
2	5.72E+17	RT @colonelkickhead: Another bloody instant restaurant week?!?! Seriously! They just jumped the shark riding two other sharks powered by shÃ¢â‚¬Â¦none
3	5.61E+17	@azzamalirhabi @JihadiA8 This video of the Peshmerga decimating ISIS is far more interesting. https://t.co/d36g1z12NPnone
4	5.76E+17	Oh really? No more instant restaurants? THAT'S SHOCKING. #MKR #MKR2015none
5	5.52E+17	RT @Benfrancisallen: It hasn't been a good few weeks for #ISIS. A new front has opened up in #Sinjar and they're about to lose the battle fÃ¢â‚¬Â¦none
6	5.63E+17	RT @NoToFeminism: I donÃ¢â‚¬â„¢t need femisnsn because men carry heavy things that i cannot!!! like shopping boxes and a huge sense of superioriÃ¢â‚¬Â¦none
7	5.63E+17	@MariachiMacabre 19% is not the vast majoritynone
8	5.72E+17	@DianH4 @ExposeFalsehood And it is Muslims who were the first crusaders attacking the Christian world for centuries before it attacked backracism
9	5.67E+17	@truaemusic @mattybboi83 @Number10gov Capital Hill is a great example of how seldom the world attacks Islam given the daily provocations.racism
10	5.65E+17	RT @fruitondabottom: #FeminismisEQUALITYwhen Men are actually listened to and part of the dialog. #HeForShe #WomenAgainstFeminism http://t.Ã¢â‚¬Â¦sexism
11	5.63E+17	Gather round kids. It's story time brought to you by my good friend @jcmanous.none
12	5.71E+17	Did you hear that? That's the sound of a bunch of women in tech setting up filters to fwd emails from @ninaburleigh to /dev/null. Good job.none

Fig 4: Raw Tweets Data Before Preprocessing

2. Pre-Processing and preparation

Step-1

We preprocessed the Data to extract the labels from raw data by splitting the tweet column and created an updated new dataset which has extra label column, and we removed tweet id column which is of no use.

```
fields = ['tweet', 'label']

with open('/content/drive/MyDrive/ ARKA/Updated_Dataset.csv', 'w', encoding='utf-8') as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow(fields)
    csvwriter.writerows(rows)
```

Fig 5: Data Preprocessing Code snippet

	tweet	label
1	RT @colonelkickhead: Another bloody instant restaurant week?!?! Seriously! They just jumped the shark riding two other sharks powered by shÃ¢â‚¬Â¦	none
2	@azzamalirhabi @JihadiA8 This video of the Peshmerga decimating ISIS is far more interesting. https://t.co/d36g1z12NP	none
3	Oh really? No more instant restaurants? THAT'S SHOCKING. #MKR #MKR2015	none
4	RT @Benfrancisallen: It hasn't been a good few weeks for #ISIS. A new front has opened up in #Sinjar and they're about to lose the battle fÃ¢â‚¬Â¦	none
5	RT @NoToFeminism: I donÃ¢â‚¬â„¢t need femisnsn because men carry heavy things that i cannot!!! like shopping boxes and a huge sense of superior	none
6	@MariachiMacabre 19% is not the vast majority	none
7	@DianH4 @ExposeFalsehood And it is Muslims who were the first crusaders attacking the Christian world for centuries before it attacked back	racism
8	@truaemusic @mattybboi83 @Number10gov Capital Hill is a great example of how seldom the world attacks Islam given the daily provocations.	racism
9	RT @fruitondabottom: #FeminismisEQUALITYwhen Men are actually listened to and part of the dialog. #HeForShe #WomenAgainstFeminism http://t.	sexism
10	Gather round kids. It's story time brought to you by my good friend @jcmanous.	none

Fig 6: Updated Dataset

Step-2:

After getting updated dataset we used LabelBinarizer class in scikit-learn library for transforming non-numerical categorical labels into numerical categorical labels.

Step-3:

After that data split into test train with test size of ‘0.20’ and Random state of ‘42’

Step-4:

We used Tokenizer to text into a sequence of tokens which can be further passed to the model for training

Tokenizer

```
[ ]  tokenizer = Tokenizer(num_words=10000)
     tokenizer.fit_on_texts(X)
     print(X_train)

     ['RT AsheSchow look at how Clinton accusers would be treated under today sexual assault s
```

```
[ ]  X_train = tokenizer.texts_to_sequences(X_train)
     X_test = tokenizer.texts_to_sequences(X_test)
     print(X_train)

     [[8, 2632, 149, 44, 55, 3015, 70, 21, 1464, 336, 244, 841, 2243, 2154, 339, 13, 12, 104],
```

Fig 6: Tokenization of the data

3. Model implementation

The implemented model follows a comprehensive approach for cyberbullying detection and classification. The model is built using the Keras library in Python, leveraging deep learning techniques such as Bidirectional LSTM layers and word embeddings. The dataset

used for training and evaluation consists of a collection of tweets labeled with three classes: cyberbullying, non-cyberbullying, and ambiguous. Preprocessing steps are applied to the text data, including removing HTML tags, non-alphabetic characters, and unnecessary white spaces. The text is tokenized and transformed into sequences, which are then padded to ensure uniform length for input into the model. The GloVe word embeddings are utilized to capture semantic information and enhance the model's understanding of the tweet content. The model architecture consists of multiple Bidirectional LSTM layers followed by dense layers for classification. The model is trained using the Adam optimizer with categorical cross-entropy loss, and performance metrics such as accuracy, precision, recall, and F1-score are measured during the training process.

The implemented model demonstrates promising results in cyberbullying detection and classification. During the training process, the model shows steady improvement in performance metrics on both the training and validation sets. The accuracy, precision, recall, and F1-score are tracked and visualized using matplotlib to monitor the model's progress. The trained model achieves a high level of accuracy and exhibits balanced precision and recall values, indicating its ability to effectively distinguish between cyberbullying and non-cyberbullying content. To ensure reproducibility, the trained model, along with the encoder classes for label transformation, is saved for future use. Additionally, a tokenizer object is serialized to facilitate text preprocessing on new data. In an evaluation phase, the saved model is loaded, and new text data is preprocessed and transformed using the tokenizer. The model then predicts the class label of the input text, providing a reliable means for real-time cyberbullying detection and classification.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 100)          2433200

 bidirectional (Bidirectiona  (None, 100, 100)         60400
 l)

 bidirectional_1 (Bidirectio  (None, 100, 108)         66960
 nal)

 bidirectional_2 (Bidirectio  (None, 120)              81120
 nal)

 dense (Dense)               (None, 64)                7744

 dense_1 (Dense)             (None, 3)                 195

=================================================================
Total params: 2,649,619
Trainable params: 216,419
Non-trainable params: 2,433,200
_________________________________________________________________
```

Fig 8: Model Summary

4. Deployment:

We implemented the Gradio application for cyber bullying detection using a trained model. The application allows users to input a text and receive a prediction of the text's class, indicating whether it contains cyber bullying or not.

The application starts by loading the pre-trained model and necessary resources, including the model itself (Cyber_Bullying_Model.model), the class labels (classes.npy), and the tokenizer (tokenIzer.pickle). These resources are essential for making accurate predictions.

The implementation of this Gradio application showcases the practical application of your trained model for real-time cyber bullying detection. It provides a user-friendly interface that can be used by individuals, organizations, or social media platforms to identify and handle instances of cyber bullying effectively.

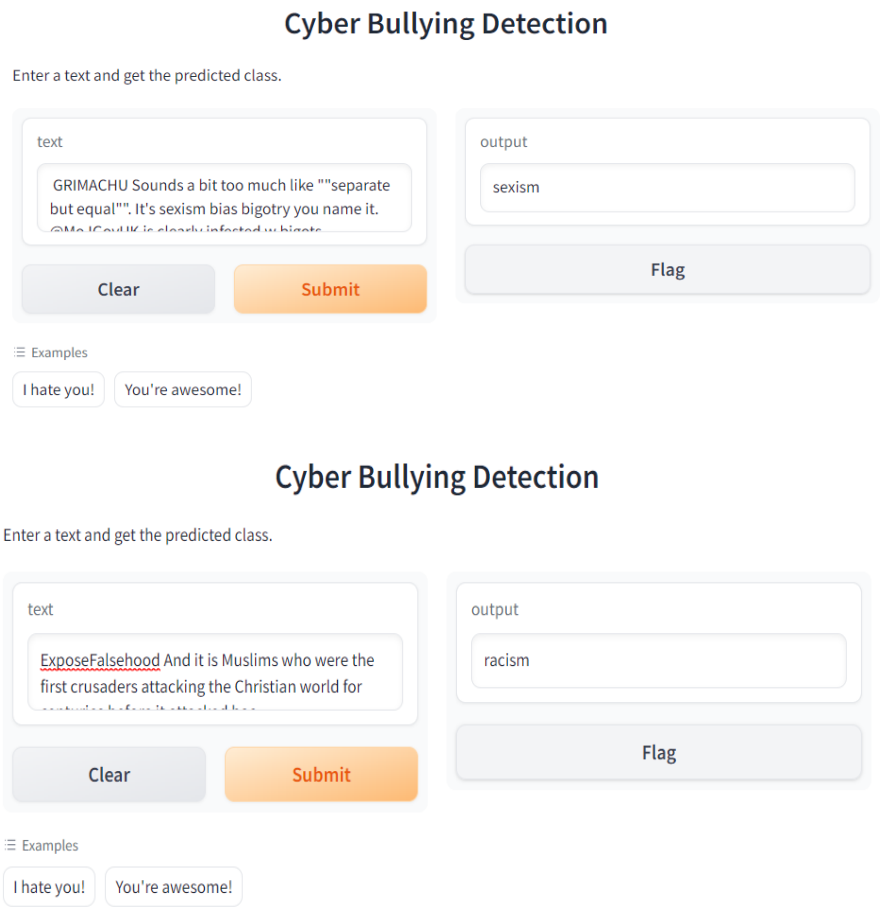


Fig 9: Gradio UI

Results:

We used 16 Epoch to train the model and Adam optimizer as optimization function and cross entropy for loss calculation.

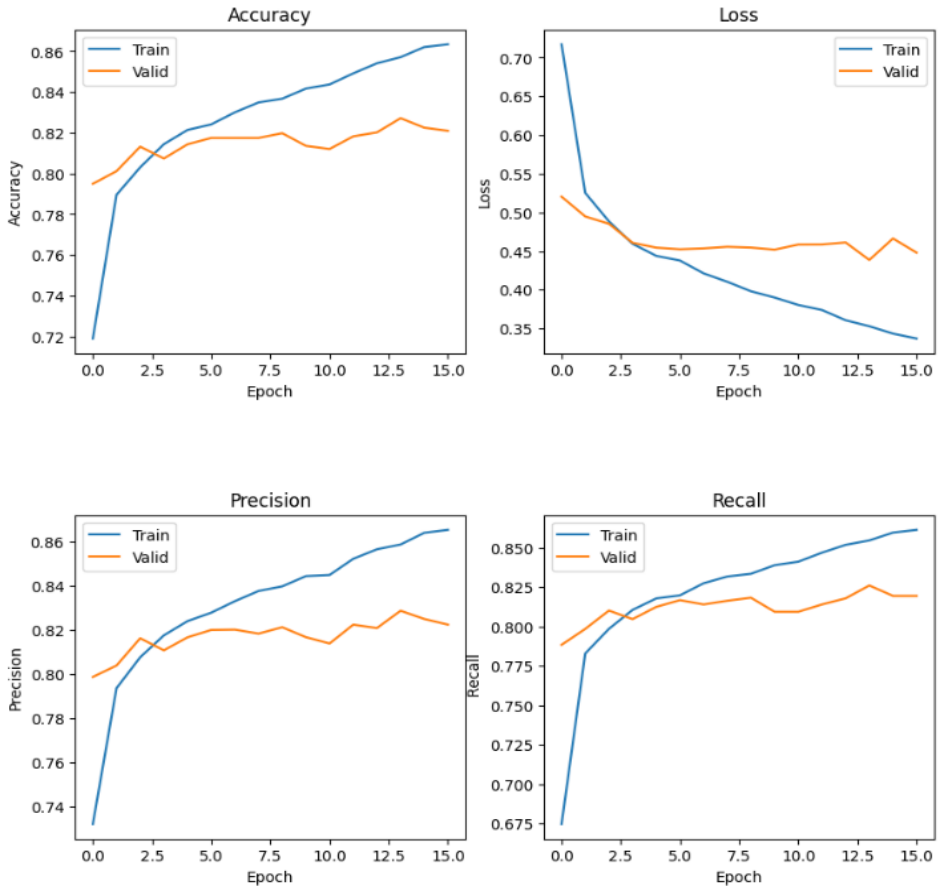
We evaluated our model using accuracy, precision, recall and f1 score metric.

```
model = Sequential([
    Embedding(vocab_size, 100, weights=[embedding_matrix], input_length=maxlen , trainable=False),
    Bidirectional(LSTM(50, dropout=0.2, recurrent_dropout=0.2, return_sequences=True)),
    Bidirectional(LSTM(54, dropout=0.3, recurrent_dropout=0.3, return_sequences=True)),
    Bidirectional(LSTM(60, dropout=0.3, recurrent_dropout=0.3)),
    Dense(64, activation="relu"),
    Dense(3, activation="softmax")])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy', Precision(), Recall(), f1_score])

model.summary()
```

Fig 10: Model



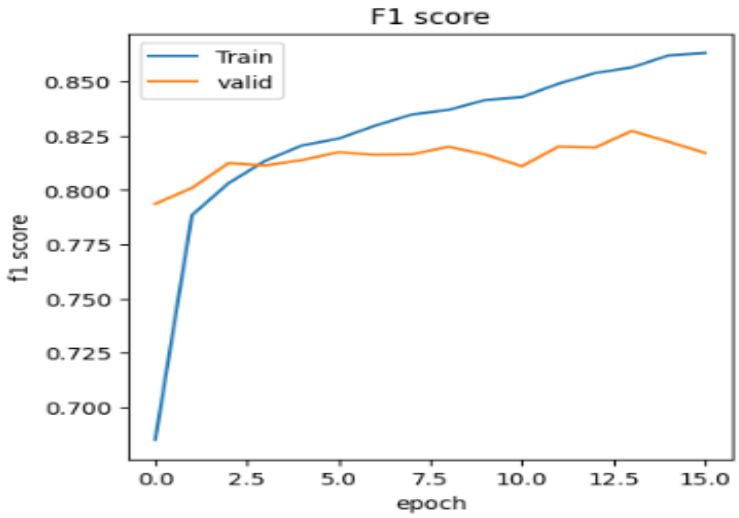


Fig 11,12: Plots

Accuracy	loss	Precision	Recall	F1_score
0.8331	0.4365	0.8337	0.8321	0.8334

Conclusion:

Cyberbullying detection is the process of locating instances of this type of bullying, which takes place in the online world. Using sophisticated technology like natural language processing (NLP) to evaluate the text and find inappropriate or abusive words is necessary for automated detection of cyberbullying. Given that cyberbullying is a widespread issue on social media and other online platforms, cyberbullying detection techniques are becoming more and more crucial. By identifying and reporting instances of abusive behavior, these technologies can assist in preventing and reducing the detrimental impacts of cyberbullying. Cyberbullying detection methods like deep learning-based models using BLSTM architecture and GloVe embedding have showed promising results. To ensure that everyone can use the internet safely and respectfully, cyberbullying detection was developed.

REFERENCES:

1. NLP and Machine Learning Techniques for Detecting Insulting Comments on Social Networking Platforms https://ieeexplore-ieee-org.unh proxy01.newhaven.edu/document/8441728
2. LSTM with working memory https://ieeexplore.ieee.org/document/7965940
3. "GloVe: Global Vectors for Word Representation" Authors: Jeffrey Pennington, Richard Socher, Christopher D. Manning Published: 2014: https://nlp.stanford.edu/pubs/glove.pdf
4. Cyberbullying detection: advanced preprocessing techniques & deep learning architecture https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00550-7
5. Accurate Cyberbullying Detection and Prevention on Social Media https://www.sciencedirect.com/science/article/pii/S1877050921002507
6. An Application to Detect Cyberbullying Using Machine Learning and Deep Learning Techniques https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9321314/

Our Project GitHub Repo:

https://github.com/agani3-UNH-DSCI/DSCI6004_NLP_FinalProject_Team_ARKA/tree/main