International Journal of Information Technology (IJIT)
Volume 4, Issue 01, January-June 2023, pp. 31-48 Article ID: IJIT_4_01_004
Available online at https://iaeme.com/Home/issue/IJIT?Volume=4&Issue=1
Journal ID: 4573-3410, DOI: https://doi.org/10.17605/OSF.IO/HMBD3

DEEP LEARNING PLUGIN TOOL USING AN HYBRID RNN-LSTM MODEL TO DETECT AESOPIAN PHRASES IN CYBERBULLYING

Vasanthan Ramakrishnan
Chief Scientist, Center for Innovation in Emerging Technologies (CIET), Chicago, USA

Lestine Grace Saquilabon
Research Scientist, Center for Innovation in Emerging Technologies, Chicago, USA

Abinesh Lingeswaran
Machine Learning Intern, Center for Innovation in Emerging Technologies, Chicago, USA

ABSTRACT

Cyberbullying is a major issue on social media that can cause significant harm to children and teenagers' mental and physical health, in some cases leading to self-harm or suicide. Existing cyberbullying detection systems primarily rely on content moderators built and managed by social media platforms. However, they do not consider complex language features such as slang, multilingualism, or aesopian phrases, which are essential in detecting cyberbullying accurately. This research proposes the world's first AI-based plug-in tool that uses a hybrid RNN-LSTM neural network to automatically detect, anticipate, and classify cyberbullying/prospective incidents while also recognizing false positives. The tool considers patterns across natural language, slang terms, and aesopian phrases that are unique to select groups or communities. The proposed tool can work across languages, browsers, devices, and community-specific phrases to protect children worldwide. The study demonstrates that deep learning techniques can be effectively used for cyberbullying detection, and the proposed tool can have the potential to save over a billion kids on the internet and social media.

Keywords: Cyberbullying, Deep learning, RNN-LSTM, Natural Language Processing, Aesopian phrases

Cite this Article: Vasanthan Ramakrishnan, Lestine Grace Saquilabon and Abinesh Lingeswaran, Deep Learning Plugin Tool Using a Hybrid RNN-LSTM Model to Detect Aesopian Phrases in Cyberbullying, *International Journal of Information Technology (IJIT)*, 4(1), 2023, pp. 31-48.
https://iaeme.com/Home/issue/IJIT?Volume=4&Issue=1

INTRODUCTION

As cyberbullying becomes more and more frequent in social networks, advanced detection and smart classification becomes of the utmost importance. The biggest challenge with the issue of cyberbullying is that cyber bullies can quickly proliferate across platforms that have a large teenage/children audience and stay hidden for months or even years before their profiles get flagged and deactivated. Cyberbullying has been known to cause major issues to children and teenager's mental and physical health, in many cases leading to self-harm or worse death by suicide.

Majority of the currently existing systems primarily focus on cyberbullying detection through stock content moderators built and managed by social media platforms like Facebook, Instagram and TikTok, which an end-user cannot directly control nor influence. Besides, these stock content moderators are intentionally programmed to work within restrictive community guidelines and their main goal is to prevent security escalation. Because the goal of these stock tools is to prevent escalations they do very little to actually determine the root cause of the bugs/loopholes that allow stalkers, bullies and trolls to hide under plainsight.

CySafe is the world's first contextual AI-based plug-in that automatically detects, anticipates and classifies cyberbullying/prospective incidents while being smart enough to recognize false positives - sarcastic content, unintended swear language, friendly banters, etc as not bullying and is capable of tracking through patterns across natural language, slang terms, aesopian phrases (intentional language only understood by a select group/community hidden away in common phrases or symbols). An example of an Aesopian phrase used for bullying a child could look like "Blk kid, U Luk Ug-Lee" or it could be more complex in the form of emojis, like say "⚫ ?? ??" . Current systems have not even begun to consider Aesopian phrasing, multi-lingual or slang terms but CySafe was built to work across languages, browsers, devices, community-specific phrases, etc to keep children all over the world safe.

A detailed look at the current state-of-the-art in cyberbullying detection reveals that deep learning techniques have rarely been used to solve this problem, despite growing reputation in other text-based classification tasks. The goal of CySafe was to develop an AI based SaaS tool for Cyberbullying Detection and Prevention, using a customized RNN-LSTM Neural Network that can have the potential to save over a billion kids on the internet and social media.

Background

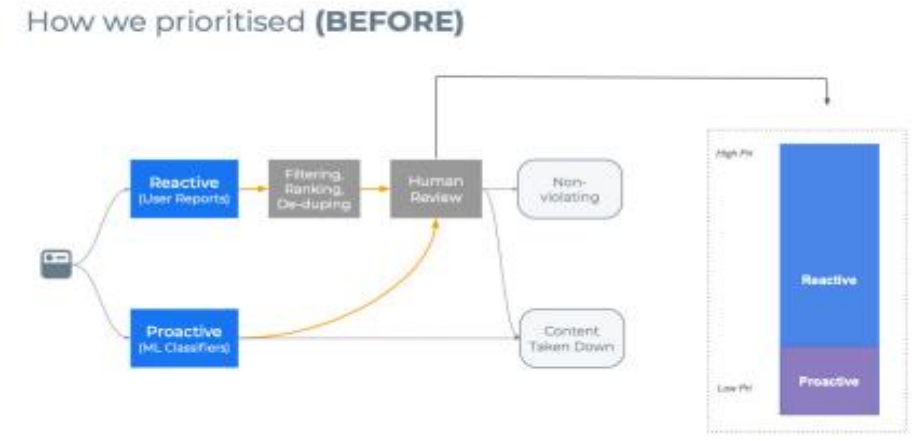

Figure 1 Picture depicting priority and work flow of a stock moderator used by a major platform

Majority of the previous work has only considered abusive language/insult words as the main characteristics to identify cyberbullying. But practically, Cyberbullying cannot be detected by considering only one characteristic. It is difficult and different from just detecting aggressive content.

Key issues with the above model,

- The model does not consider locality/region and may be potentially missing out on slang and/or community-specific multilingual terms that could be harmful
- The model does not appear to employ Natural Language Processing (NLP) or an advanced Deep Learning technique that puts it at the risk of missing incomplete sentences, intentional typos and multimedia texts with emojis and aesopian phrases that could still be harmful
- Classic Homoscedasticity error: if different sampling groups being compared are heterogeneous, the above model will fail
- Human review only exists for a certain condition, trained programmers can learn then triggers to avoid detection

Assuming if the above model is used, the classifier would have considered calling someone stupid, idiot as cyberbullying, but in a scenario where the same words are exchanged in friendly banter it would trigger a false positive and end up flagging the user when it is not necessary to do so. In the current social media platforms, there are more extreme words, complex ways and nuances to harass/bully someone and existing systems just are not using the right tools to accomplish the task. In order to accurately detect Cyberbullying, a system capable of adapting to human languages must be employed.

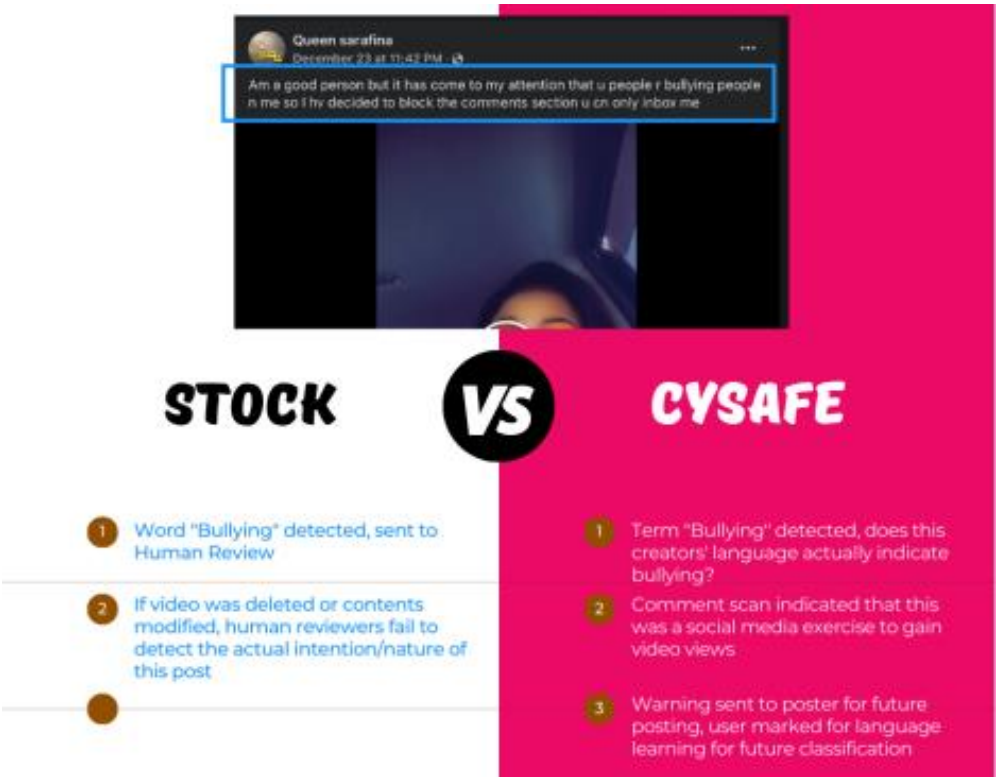

Figure 2 shows a visual comparison of stock moderators and CySafe depicting the unique features the latter has over the former

CySafe also helps cyberbullying victims and prospective victims to take the incident to their control and make a decision on whether or not the incident or encounter was actually bullying or not which gets fed into the system to train the Deep Learning model.

Methods

Concepts Involved:

The following concepts stated below were used for implementing the algorithm

1. Natural Language Processing

NLP is a subfield of computer science and artificial intelligence concerned with interactions between computers and human (natural) languages. It is used to apply machine learning algorithms to text and speech.

Basics of NLP:

- Sentence Tokenization: Converting para to sentences
- Word Tokenization: Converting sentences to words
- Text Lemmatization: Verifying meaning of the word and returning the base meaning
  Ex: better, best => good
- Stemming: Removal of derivational affixes.
  - Ex: playing, play, playful => play
  - Stop Words: Filtering out the commonly used words like the, a, an, in, on, etc…
  - Regex: Matching words based on a regular expression pattern sequence
  - Bag-of-Words: It describes the occurrence of each word within a document.

- Term Frequency (TF): a scoring of the frequency of the word in the current document.

Term Frequency Formula

$$TF(\boldsymbol{term}) = \frac{Number\ of\ times\ \boldsymbol{term}\ appears\ in\ a\ document}{Total\ number\ of\ items\ in\ the\ document}$$

- Inverse Term Frequency (ITF): a scoring of how rare the word is across documents.

Inverse Document Frequency Formula

$$IDF(\boldsymbol{term}) = \log\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ \boldsymbol{term}\ in\ it}\right)$$

TF-IDF: We penalize words that are frequent across all the documents. This approach is called TF-IDF. TF-IDF, short for term frequency-inverse document frequency is a statistical measure used to evaluate the importance of a word to a document in a collection.
$TFIDF(\boldsymbol{term}) = TF(\boldsymbol{term}) * \mathrm{IDF}(\mathbf{term})$

- Part-of-speech (POS) tagging: It is a process of converting a sentence to forms – list of words, list of tuples (where each tuple is having a form (word, tag)). The tag in case of is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on.

2. Recurrent Neural Networks

Recurrent Neural Network (RNN) is a type of Neural Network where the output from previous step is fed as input to the current step.

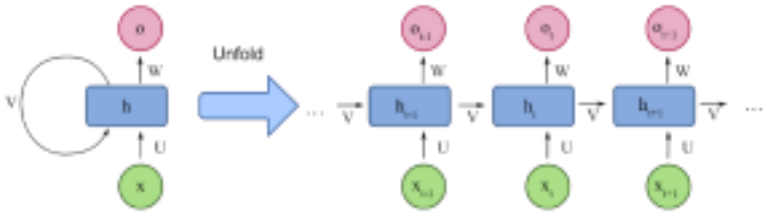

Figure 3 shows a visual working of a typical Recurrent Neural Network (RNN)

Limitations of RNN:

RNN are also not very good in capturing long term dependencies and **the problem of vanishing gradients** resurface in RNN.

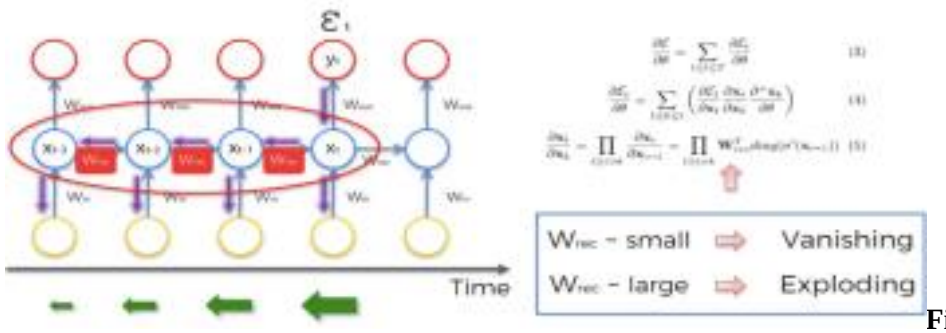

Figure 4 shows a picture depicting the output of an RNN model highlighting its limitation

3. LSTM

Long Short-Term Memory networks are the special mode of Recurrent Neural Networks (RNN) that have the capabilities to learn long-term dependencies.

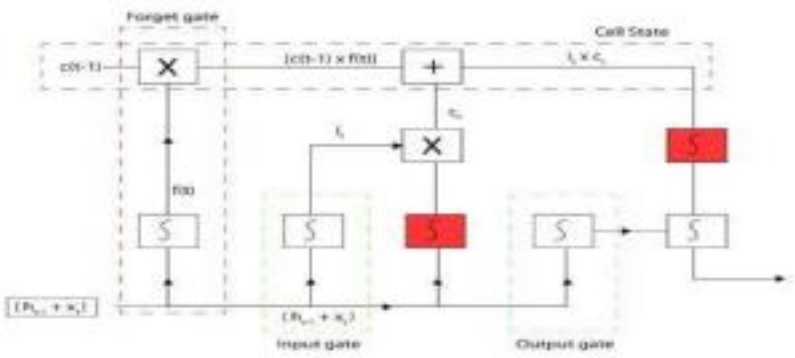

Figure 5 shows the components of an LSTM model

Forget Gate:

The first block represented in the LSTM architecture is the forget gate (ft). The information from the current input (Xt) and the previous hidden state (ht) is passed through the sigmoid activation function. If the output value is closer to 0 means forget, and the closer to 1 means to retain.

Input Gate:

This gate is processing the (ht-1 + xt) and giving out new input using the activation function which is usually Sigmoid activation function, again in the range of 0 to 1, not ignoring any information like forget gate.

The new generated input is known as the gate input (it). Parallely (ct) is being processed under the Tanh activation function which generates the output in the range of -1 to 1.

Finally, multiply the tanh output (C't) with the sigmoid output (it) to decide which information is important to update the cell state.

Cell State:

The input from the previous cell state (Ct-1) is pointwise multiplied with the forget gate output. If the forget output is 0, then it will discard the previous cell output (Ct-1).

This output is pointwise added with the input gate output to update the new cell state (Ct).

$$\text{Cell state} = (ct\text{-}1) * (ft) + (ct) * (it)$$

Output Gate:

Output gate processes (ht-1 + xt) under the Sigmoid Activation function which squeezes the information in the range of 0 to 1, is further multiplied with the cell state information.

There are two types of LSTM:

1) Unidirectional LSTM: In this case the LSTM model saves information from the past when the model reads through the input word by word. So in this case the information passes in only one direction

2) Bi-directional LSTM: This model makes the input pass in both directions, one is from past to future and then future to past. So when the model runs backwards it has preserved information from the future and using the hidden states it is possible to preserve information from both past and future and use it when needed. This is specifically used for sentence classification and entity recognition. So Bi-directional LSTM will have more trainable parameters that normal LSTM and so it takes longer to train the model.

4) Text Data Augmentation

Data Augmentation is a process to artificially increase the size of the given data set by generating various versions of the real datasets present. This is done without actually collecting any new data. Data Augmentation in Natural Language Processing and Deep Learning is used to deal with the problems of data scarcity and less data diversity.

When the data is augmented it should be made sure that the augmented data is neither very similar or very different from the original dataset, since this may lead to overfitting or poor performance in accuracy using validation data. There are basically 3 different types of text augmentation:

1) Easy Data Augmentation:

In this method, any random word in the sentence is selected and replaced with one or more synonyms or any two words in the sentence are swapped.

1.1) Synonym Replacement:

Contextual Bi-directional embedding like BERT is used for richer vector representation. Since Bi-directional LSTM and Transformer based models encode longer text sequences and have contextual knowledge of their surrounding words.

1.2) Random Insertion:

Inserting synonyms of randomly selected words at different positions in the sentence and this word is not among stopwords.

2) Back translation:

Since English is a very common language, it has lots of training data for translation, while some languages may not have enough data to train a machine translation model. BackTranslation translates target language to source language and combines both original source and back-translated sentence to train the model.

3) Generative Models:

Generative adversarial networks (GAN) is a technique used for cross-domain adaptation. This model is trained to generate text using a few words and generative language models like BERT, BART, T5 are used to generate text in a more class category preserving matter. This model encodes the class category along with its associated text sequences to generate newer samples with some modifications.

Implementation

The goal of this report is to design and train a hybrid RNN-LSTM model and test it on a given dataset. The core development team uses a sample YouTube dataset which consists of comment section texts, to validate the model.

The dataset was used to create 3 different models:

1) Using Natural Language Processing (NLP)

As the first step, we implement the NLP-based classification model to classify whether the given statement is an indication of Cyberbullying or Not. In this algorithm:

1.1) The given data is extracted from both the training and testing data CSV file and stored in 2 different panda data frames. Since all the tasks given, must be applied to both the testing and training data (later used to create the model) they both are concatenated and stored

1.2) Now, the process of tokenization is applied, where every sentence is split into chunk of words

1.3) After this, now the given data is checked for presence of noun, verb or adjective etc. by using WordNetLemmatizer. Now, the stop words which were found using Lemmatization are removed.

1.4) After stop words removal, the Term Frequency-Inverse Document Frequency (TF-IDF) is implemented before using these features to train the model. TF-IDF is to numerical value to find how important a word is to the collection of data

1.5) Now this model is trained using the Random Forest Ensemble Classifier

2) Using RNN-LSTM (Long-Short Term Memory) neural network architecture

The next step was to design a cyberbullying detection using a LSTM Deep Learning Neural Network. They were implemented in 2 different models:

1. **LSTM**
2. **Bi-Directional LSTM**

We implement both the LSTM and bi-directional Long-Short Term Memory (LSTM) Neural Network using the TensorFlow deep learning network library for the given YouTube dataset. We then execute the model using a binary cross entropy loss and used an ADAM optimization algorithm to train the deep learning model.

3) Implementation of RNN-LSTM after augmenting the Data:

In this project there are two main methods used for augmentation of the YouTube Dataset:

3.1) WordNet Augmenter

Wordnet augments text by replacing words with synonyms provided by WordNet.

3.2) Bi-Directional Encoders Representation from transformers (BERT)

BERT is a transformer model pre-trained on raw texts of data only, without any labeling on them. It is an automated process to generate inputs and labels from those texts. There are 2 usages for this model:

1) Masked-Language Modelling (MLM) and Next Sentence Prediction.

Since it's trained on non-labelled data it naturally learns an inner representation of the language, which can then be used to extract features. After augmenting the dataset increased by 4501 datasets. Initially the size of the dataset was 20000 and after augmenting the dataset increased to 24501

Results And Observations:

1) Result for the NLP machine learning model implementation:

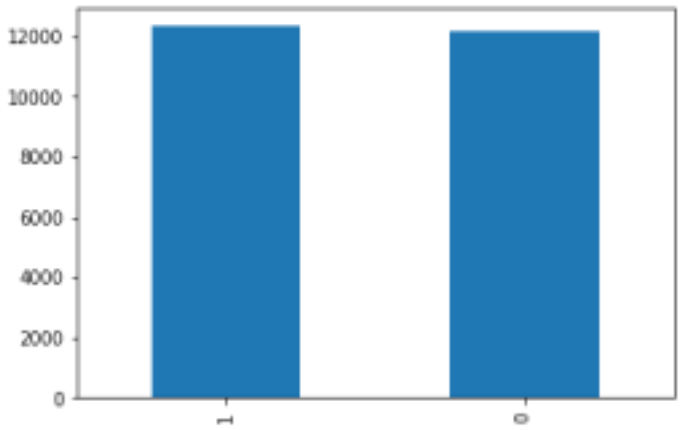

Figure 6 shows the results of the NLP ML model implementation

The above diagram represents the bar plot of classified labels, where "1" – Bullying detected, "0"- No Bullying. From the graph, we can conclude that in a total dataset of size 24501 statements:

12,708 are bully statements ("1")

11,793 are no bully statements ("0")

	Precision	recall	f1-score
0	0.70	0.90	0.79
1	0.72	0.41	0.52
accuracy	0.71		
macro avg	0.71	0.65	0.66
weighted avg	0.71	0.71	0.68

It can be seen that the accuracy of the model is 71%. Ideally, the model is considered good if the accuracy is between 70% - 90%. With more data, the model has the potential to improve its accuracy.

2) Result for implementation of hybrid RNN-LSTM Model:

The learning curve plots (accuracy vs epoch) and (loss vs epoch) is plotted below for all the models tested. Note that in all the plots the training curve remains the same, since all models are trained on the same dataset. The training accuracy curve gradually increases and flattens/saturates at the 14th epoch and the training loss curve gradually decreases as the epochs/training data increases. Using the validation curve in accuracy and loss we can predict the most suitable model.

2.1) Uni-Directional LSTM

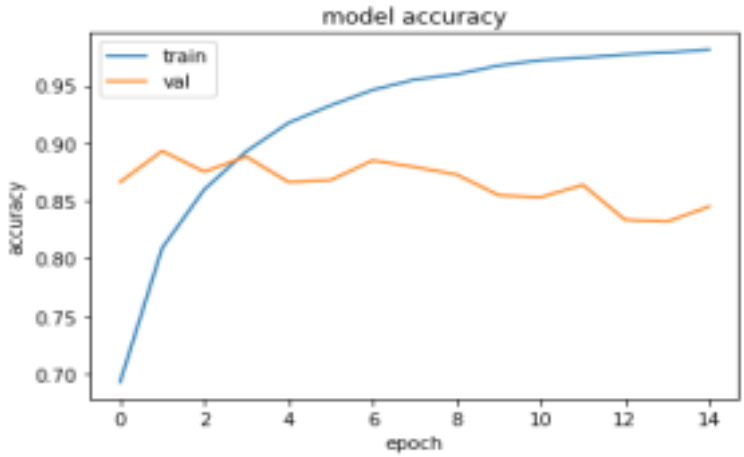

Figure 7 shows a graph with validation accuracy for the uni-directional LSTM model

The validation accuracy curve starts at a high value and fluctuates and progressively decreases

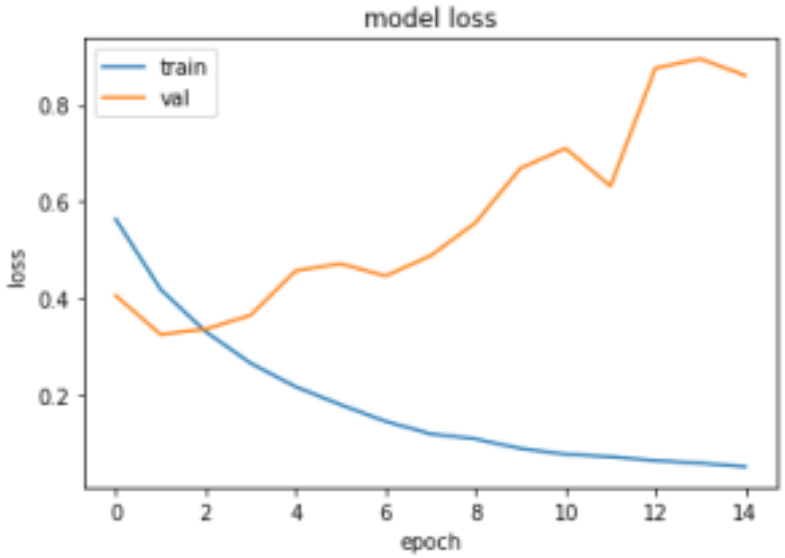

Figure 8 shows the training and validation loss over time

The training and validation loss are close to each other initially and progressively increases as the datasets/epochs increase

2.2) Bi-Directional LSTM:

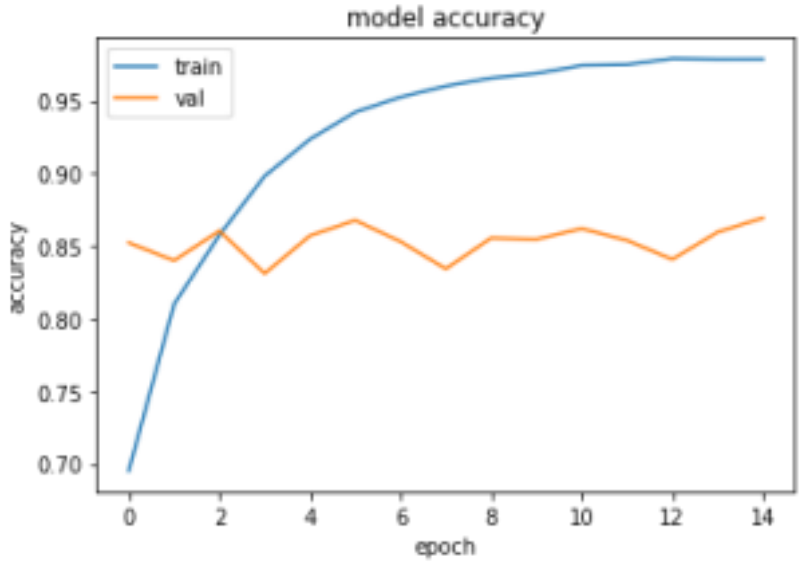

Figure 9 shows the model accuracy for the bi-directional LSTM model

The validation accuracy curve starts at a high value and fluctuates and progressively flattens

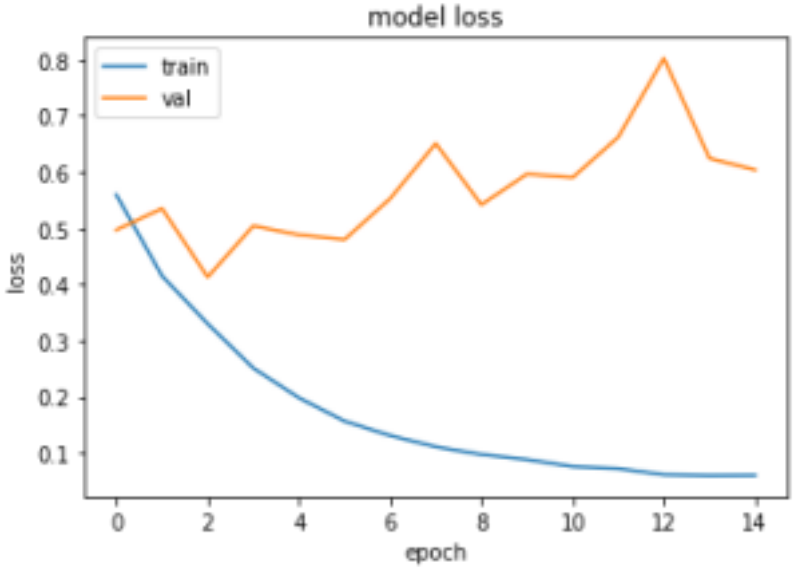

Figure 10 shows the model loss over time

The training and validation loss are close to each other initially and progressively increases as the datasets/epochs increase

3) Result for implementing after data augmentation:

3.1) Augmentation using BERT

3.1.1) Uni-Directional LSTM

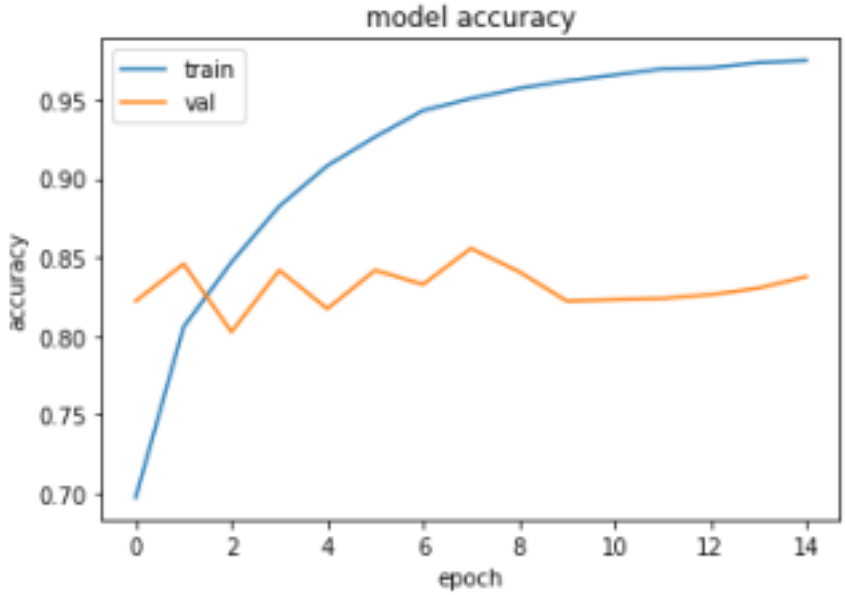

Figure 11 shows the model accuracy after data augmentation for the unidirectional model

The validation accuracy curve starts at a high value and fluctuates and progressively flattens

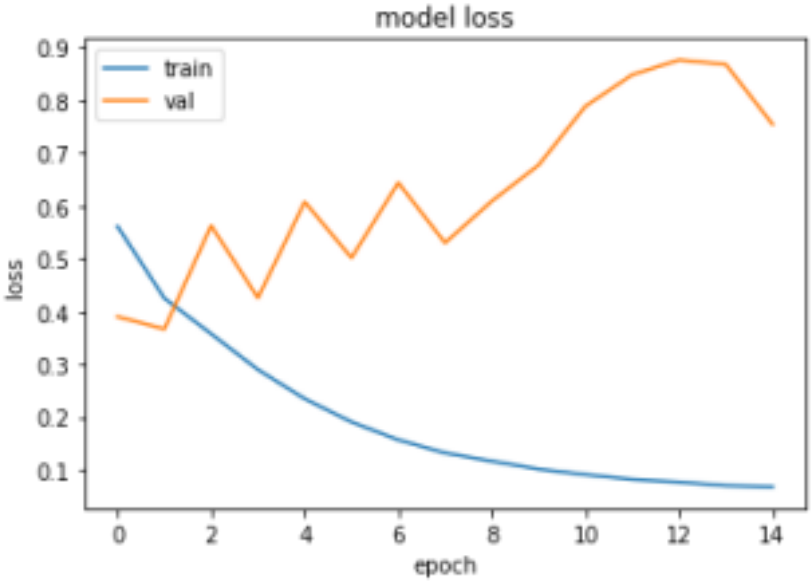

Figure 12 shows the model loss fluctuation

The training and validation loss are close to each other initially and progressively increases as the datasets/epochs increase.

3.1.1) Bi-Directional LSTM

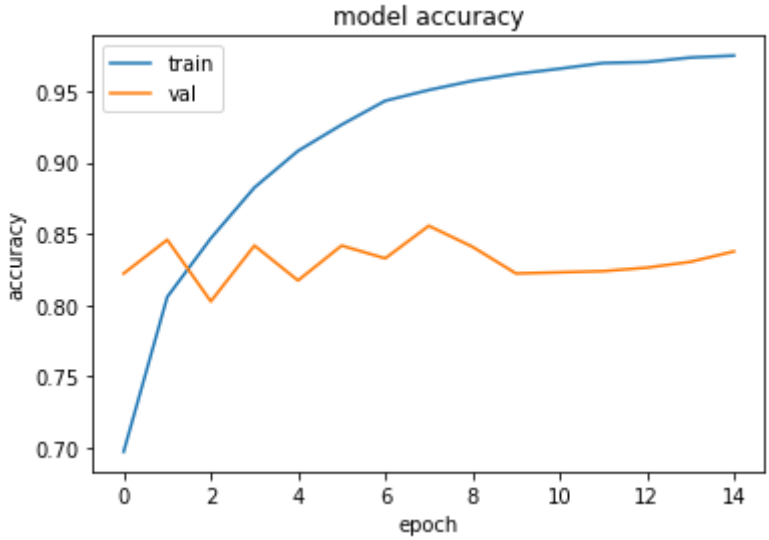

Figure 13 shows the model accuracy for the bi-directional LSTM after augmentation

The validation accuracy curve starts at a high value and fluctuates and progressively flattens but it does not perform good compared to the unidirectional since the accuracy is decreasing.

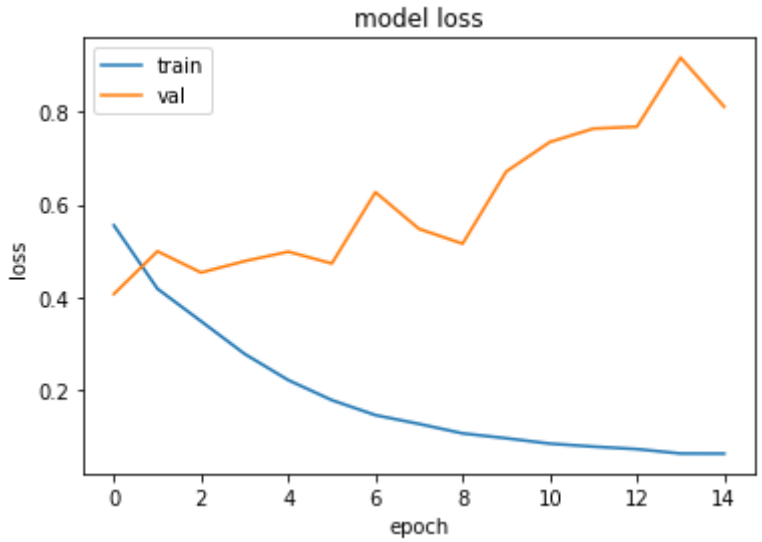

Figure 14 shows the model loss again for the bidirectional LSTM model

The training and validation loss are close to each other initially and progressively increases as the datasets/epochs increase.

3.2) Augmentation using WordNet

3.2.1) Uni-Directional LSTM

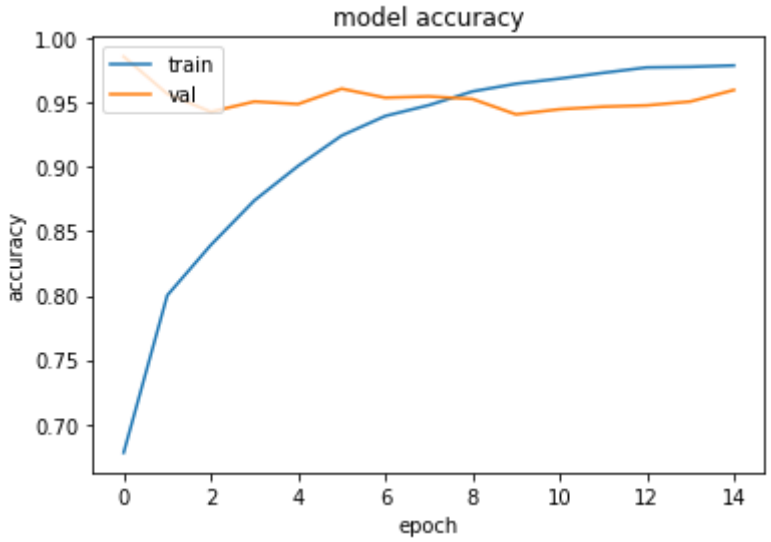

Figure 15 shows the model accuracy for unidirectional LSTM with WordNet augmentation

It can be seen that the difference in accuracy between training and validation starts high and gradually reduces and flattens after the 8th epoch.

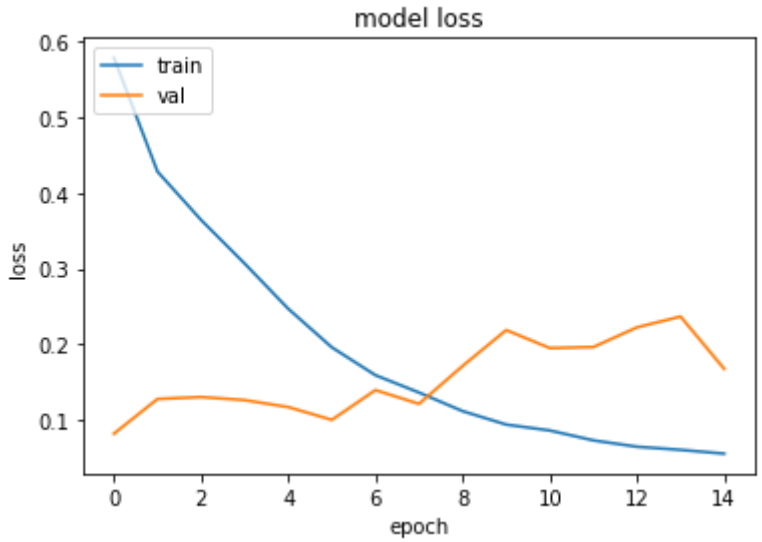

Figure 16 shows the model loss flattening for the same model

The validation loss starts at less value and slowly the difference in loss values reduces and flattens

3.2.2) Bi-Directional LSTM

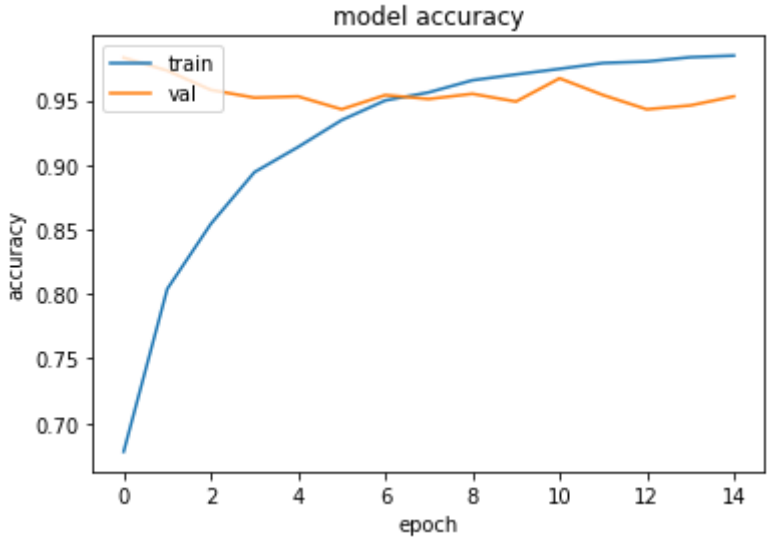

Figure 17 shows the model accuracy for the bi-directional LSTM with WordNet augmentation

It can be seen that the difference in accuracy between training and validation starts high and gradually reduces and flattens after the 6^{th} epoch.

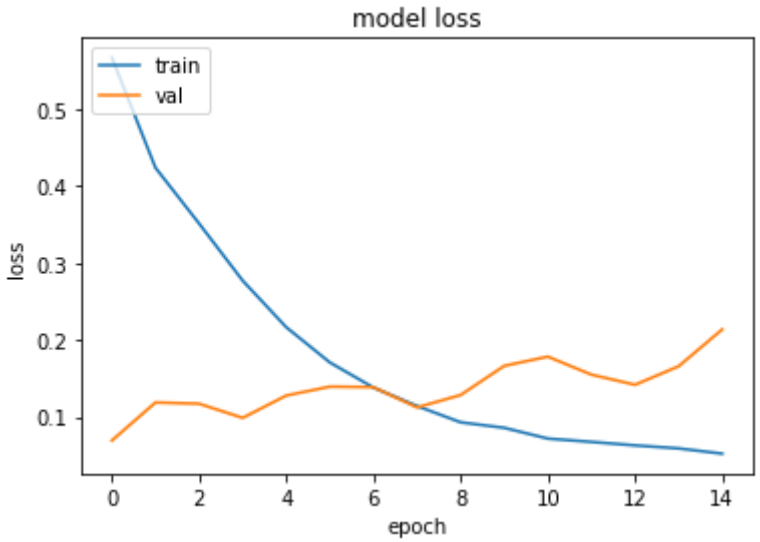

Figure 18 shows the model loss flattening for the same model

The validation loss starts at less value and slowly the difference in loss values reduces and flattens

CONCLUSIONS

As seen from the results, initially when implementing with the given dataset, the training loss is less than validation loss and it progressively increases as the epoch's increase, which shows the model is slightly overfit.

When the Augment data using WordNet is used to train the model, it performs much better than the previous one. It can be noted that the validation accuracy is higher and is more stable and the difference in validation loss and training loss is less compared to non-augmented dataset.

It can still be noted that the validation loss is more than training loss which shows that we need more data to improve the performance of the model. So, the results show that Data Augmentation is a method to solve overfitting.

Detection of a text message which comes under the category of cyber bullying would be quite challenging, as the chats containing swear words are nowadays common among the individual in this modern society. Rather than the traditional BOW, TF-IDF model, we can expect better results by using a LSTM model as the order at which words appear matters in these kinds of classification problems.

Considering the results, we can conclude that the implementation of RNN-LSTM model with augmented data produced better results and the model is ready to be implemented into a core plug-in development.

Pseudo Code

Data Preparation:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from                    sklearn.metrics                    import
accuracy_score,confusion_matrix,classifi cation_report
from sklearn.pipeline import Pipeline
from   sklearn.feature_extraction.text   import   TfidfVectorizer
X_train,X_test,y_train,y_test=train_test_split(data['content'],dat
a[
'label'],test_size=0.3,random_state=0,shuffle=True,stratify=data['
la bel'])
```

LSTM Implementation:

```
def ml_model(model, df, name = "lstm"):
  voc_size = 5000
onehot_repr=[one_hot(words,voc_size)for words in df['content']]
   sent_length=20
  embedded_docs=pad_sequences(onehot_repr,padding='pre',maxlen=s
ent_ length)
  X_final=np.array(embedded_docs)
  y_final=np.array(df['label'])
  history                                                    =
model.fit(X_final,y_final,validation_split=0.05,epochs=1
5,batch_size=64)
def lstm():
  embedding_vector_features=40
  model=Sequential()
  model.add(Embedding(voc_size,embedding_vector_features,input_l
engt h=sent_length))
  model.add(LSTM(100))
  model.add(Dense(1,activation='sigmoid'))
  model.compile(loss='binary_crossentropy',optimizer='adam',metr
 ics= ['accuracy'])
  print(model.summary())
  return model
ml_model(lstm(), data, "lstm")
```

Bi-Directional LSTM:

21

```
def bilstm():
  embedding_vector_features=40
  model1=Sequential()
  model1.add(Embedding(voc_size,embedding_vector_features,input_
leng th=sent_length))
  model1.add(Bidirectional(LSTM(100)))
  model1.add(Dense(1,activation='sigmoid'))
  model1.compile(loss='binary_crossentropy',optimizer='adam',met
rics =['accuracy'])
  print(model1.summary())
  return model1

ml_model(bilstm(), data, "bilstm")
```

Text Data Augmentation:

```
import nlpaug.augmenter.char as nac
import nlpaug.augmenter.word as naw
import nlpaug.augmenter.sentence as nas
import nlpaug.flow as nafc
from nlpaug.util import Action
import nltk
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
nltk.download('stopwords')
```

BERT:

```
aug = naw.ContextualWordEmbsAug(model_path='bert-base
uncased', action="substitute",aug_max=3)
```

WordNET:

```
aug = naw.SynonymAug(aug_src='wordnet',aug_max=3)
```

Implement Dataset Augmentation:

```
augmented_sentences=[]
augmented_sentences_labels=[]
```



```
for i in range(4500):
    temps=aug.augment(bully["content"][i],n=1)
    augmented_sentences.append(temps)
    augmented_sentences_labels.append(1)
dfaug =
pd.DataFrame({"content":augmented_sentences,"label":augmente
d_sentences_labels})
dfaug.head()
```

Augmented dataset model Implementation:

```
def model_testing(df):
  X_train,X_test,y_train,y_test=train_test_split(
      dfappend['content'],
      dfappend['label'],
      test_size = 0.2,
      random_state = 0,
      shuffle=True,
      stratify=df['label'])
  clf =
Pipeline([('tfidf',TfidfVectorizer()),('clf',SVC(C=1000,gamm
a='auto'))])
  clf.fit(X_train,y_train)
  y_pred=clf.predict(X_test)
  return accuracy_score(y_test,y_pred)
```

Unidirectional LSTM:

```
ml_model(lstm(), dfappend, "lstmaug")
```

Bidirectional LSTM:

```
ml_model(bilstm(), dfappend, "bilstmaug")
```

REFERENCE

[1] Ashwini Manish Brahme, Shivaji Mundhe, Ashwini Chavan, Sunil B Joshi and Poonam Sawant, 2013. A Review of Cyberbullying and Cyber Threats in Education. International Journal of Computer Engineering and Technology (IJCET) -Volume:4, Issue:3, Pages:324-330.

[2] Nikhil Vijayrao Khandar and Jitendra Sheethlani, Sentiment Analysis Approach in Natural Language Processing for Data Extraction, International Journal of Computer Engineering and Technology 13(2), 2022, pp. 27-32

[3] Saima Tabasum, Imdad Hussain Siddiqui, Aqsa Muhammad Siddiqu, Aamir Iqbal Umranid, Imran Ali Mangrio, The Existence of Workplace Cyberbullying among White Collar Workers: Prevalence, Gender Structure and Organizational Position, International Journal of Management (IJM), 12(3), 2021, pp. 1020-1033

[4] Deepanshu Girsa and Sanjiv Kumar Tomar, Study on Cyberbulling Detection Over Twitter Using Machine Learning Algorithms, International Journal of Management (IJM), 11(6), 2020, pp. 2335-2346

[5] Zaizul Ab Rahman, A`Dawiyah Binti Ismail, Siti Norul Huda Sheikh Abdullah, Wan Fariza Fauzi and Nur Riza Binti Mohd Suradi, Developing Self-Identity Among Teens Towards Personal Empowerment, International Journal of Civil Engineering and Technology, 9(13), 2018, pp. 674-684

Citation: Vasanthan Ramakrishnan, Lestine Grace Saquilabon and Abinesh Lingeswaran, Deep Learning Plugin Tool Using a Hybrid RNN-LSTM Model to Detect Aesopian Phrases in Cyberbullying, International Journal of Information Technology (IJIT), 4(1), 2023, pp. 31-48

https://doi.org/10.17605/OSF.IO/WFQRN

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJIT/VOLUME_4_ISSUE_1/IJIT_4_01_004.pdf

editor@iaeme.com