Explainable Detection of Online Sexism (EDOS)

Ariyan Hossain
ID: 20101099
Dept. of CSE
BRAC University
Dhaka, Bangladesh
ariyan.hossain@g.bracu.ac.bd

Rakinul Haque
ID: 20101290
Dept. of CSE
BRAC University
Dhaka, Bangladesh
rakinul.haque@g.bracu.ac.bd

Nowreen Tarannum Rafa
ID: 20101329
Dept. of CSE
BRAC University
Dhaka, Bangladesh
nowreen.tarannum.rafa@g.bracu.ac.bd

I. Introduction

The prevalence of sexism in online texts seems to be rising in the modern era. Identifying instances of sexism in online environments is a crucial NLP challenge, as it can have an adverse impact.

Online sexism refers to the various forms of discriminatory behavior and language directed towards individuals based on their gender, on the internet. Some common examples of online sexism include misogynistic comments, derogatory remarks, and harassment. The detection of online sexism can be challenging, as it often takes the form of subtle and indirect behavior that can be difficult to identify. However, there are several approaches that can be used to detect online sexism.

Detecting online sexism requires a multifaceted approach that involves the use of automated tools, user reporting, and human moderation. By implementing these measures, online platforms can work to create a safer and more inclusive environment for all users. So, to do so, we have followed methods that include Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (biLSTM), and Gated Recurrent Unit (GRU) models.

A Convolutional Neural Network (CNN) is a type of artificial neural network commonly used for image and video processing tasks. It applies a series of convolutional filters to the input image or video to extract features and reduce the dimensionality of the data. [1] These features are then fed into fully connected layers to produce the final output. CNNs can be useful for detecting sexism in online content as they are particularly effective in processing and analyzing images, which are often used in online content such as memes and social media posts. [2] Additionally, CNNs can also be used to process natural language data, such as text, which is also commonly used in online content containing sexism, which is why we chose to work with this model.

Long Short-Term Memory (LSTM) is a type of recurrent neural network that uses memory cells and gating mechanisms to selectively remember or forget information. This enables it to model long-term dependencies in sequential data. It is widely used in natural language processing, speech recognition, and time-series prediction. A biLSTM model (Bidirectional Long Short-Term Memory) combines two LSTM networks.The input sequence is simultaneously processed in both forward and backward directions in a biLSTM network. For applications like speech recognition and natural language processing, the biLSTM network can capture not only the present context but also the context before and after each element in the sequence by processing the sequence in both directions.After that, two LSTMs' outputs are put together to create the final output sequence. This is why we have used biLSTM for its ability to analyze complex sequential data. Moreover, we used it to capture long-term dependencies, which are often necessary for understanding the context in which sexist language is used.

Similar to LSTM (Long Short-Term Memory) networks, a GRU (Gated Recurrent Unit) model is a neural network with a more straightforward design that enables quicker training and uses less computer resources. GRUs work by storing and updating details about earlier inputs in a memory cell to analyze sequential data, such as text or time series data. LSTMs make use of distinct memory and hidden states, whereas GRUs do this by utilizing a gating mechanism that selectively updates the memory cell. A GRU model has the benefit of being able to capture long-term dependencies in a phrase, which enables it to comprehend the context of a comment and the link between various words. This is crucial since sexism frequently appears subtly and might be hard to identify based just on words or phrases.

In conclusion, we have worked to effectively detect sexism in different online contents by carefully analyzing the output of CNN, biLSTM and GRU models.

II. Data Exploration

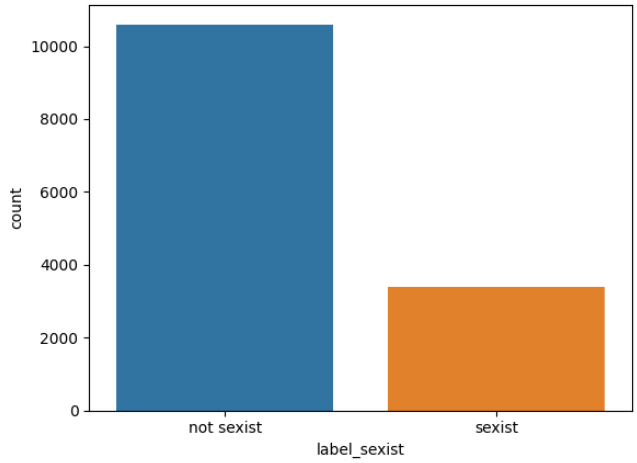
A. Data Collection

We have collected our dataset from SemEval-2023 Task 10: Explainable Detection of Online Sexism. Kirk et al., 2023 collected these data from publicly available Gab posts and subreddits and cleaned and annotated them. [3]

B. Data Observations

The dataset contained 14000 rows and 5 columns. Among the columns, there were 'rewire_id', 'text', 'label_sexist', 'label_category' and 'label_vector'. The 'text' column contained sexist or not sexist contents and the contents were labeled as sexist and non-sexist in the 'label_sexist' column. It was further classified into the category of sexism for example:

derogation, animosity etc. and the reason behind why it was sexist was also specified as label_vector. As our objective was to flag the contents as sexist and non sexist, we dropped all the columns except for 'text' and 'label_sexist column'. We then compared the sexist and non-sexist classes and clearly saw there is a class imbalance.

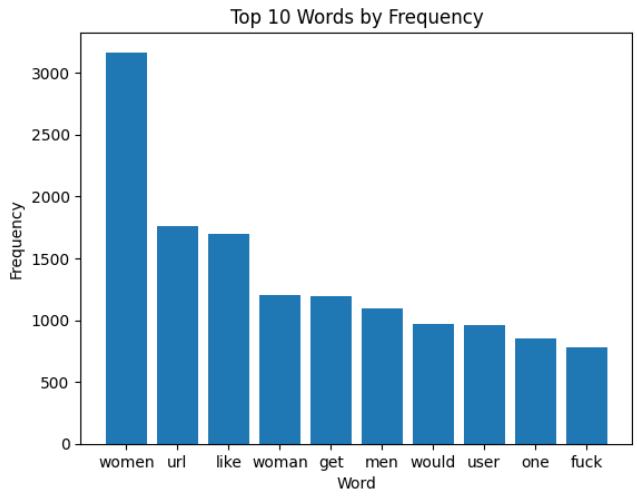


We kept the data as it is instead of balancing as it reflects real-world distribution.

C. Data Preprocessing

Before training our model, we applied a few preprocessing techniques on them to filter out unimportant parts of the sentences. We had to lowercase all the sentences. Then we removed the HTML tags and removed the punctuations and numbers. Additionally, we eliminated any single characters present which was necessary as removing punctuations left many single characters. We also removed consecutive multiple spaces in between the words. There were many emoticons and emojis present which had to be eliminated too. The cleaned texts or contents were then kept in a column named 'clean_text'.

After exploring the data, we found there are 17093 unique words in our corpus. Among those words, the most frequent words were women followed by url,like,women,men in sequential order after removing the stopwords.



Next, we labeled the sexist sentences as '1' and not sexist sentences as '0' and stored them in a column called 'label_sexist'.

D. Data Splitting

We allocated 70% data for training and 15% for validation and 15% for test set. In order to assess how well the model performed during training, a subset of the data called the validation set was employed. It is used to fine-tune the model's hyperparameters, including the learning rate or regularization strength, to boost the model's functionality. The data were divided into these subgroups to prevent overfitting, which happens when a model learns to fit training data too well and is unable to generalize to new data.

E. Preparing Embedding Layer

This layer is important because most machine learning models cannot work directly with raw data hence they need to be converted into numeric form. Tokenizer from Keras was used to tokenize the sentences and each unique word was assigned a unique integer index, based on its frequency in the corpus. Each text in the corpus was transformed into a series of numbers by the tokenizer, each of which represents a word's index in the lexicon.Padding was applied to pad the sequences in the train dataset with zeros, so that all sequences have the same length of 100 in our case. This was used to prepare our data for use in a neural network that expects inputs of fixed length. This is a common preprocessing step in natural language processing tasks, such as text classification. Lastly, we used the Glove Word embeddings file and created a 100 dimensional word embeddings dictionary of all the words found in our dataset. Since GloVe embeddings have already been trained on a sizable text corpus, they already know useful representations for a wide variety of common words. Comparing this to training embeddings from scratch on a smaller corpus can result in significant time and computing resource savings.

III. Methodology

We used Convolutional Neural Network (CNN), Bidirectional LSTM and Gated Recurrent Unit (GRU) models to perform binary classification of sexist or not sexist texts. [4]

A. Defining Model Architecture

First, we created a new sequential model object that can be used to stack layers in a neural network. Then, we created an embedding layer that converts each token in the input sequence to a fixed-length vector representation. In the parameters of the Embedding function, we passed the number of unique words in the vocabulary, dimension of the output vector which we set as 100, initial weights for the embedding layer from the embedding matrix that contained pre-trained word embeddings, maximum length of input sequence as 100, and trainable as False because we did not want the embedding layer to be updated during training. Afterwards, we added the embedding layer to the sequential model.

Then we add the respective model to the sequential model and use ReLU as the activation function for the hidden layer. ReLU was used as it is simple, helps reduce overfitting, and is non-saturating which makes it a good choice of activation function for the hidden layer. For CNN and Bi-directional LSTM, we allocated 128 memory units and 256 memory units for GRU. Lastly, we added the dense having 1 output unit as we were performing binary classification and used sigmoid for activation function. Sigmoid was used in the output layer as it converts any input value to a probability represented by a number between 0 and 1. While compiling the model, we used 'adam' as optimizer. The optimizer is used to update the weights during training. The Adam optimizer is a variant of stochastic gradient descent (SGD) that is computationally efficient and works well for classification problems. We used binary cross entropy as the loss function as it was gradient-friendly, interpretable, and widely used.

B. Model Training

During our model training, we used early stopping which is a regularization technique used during training to prevent overfitting. This method monitors the validation loss and stops training if the loss does not improve for a certain number of epochs. During training, we set a small batch size as our dataset was small and set it to 64 and set epoch to 20. This high number of epochs does not cause any problem because of early stopping. While training, the weights of the model are updated based on the input data and target labels, using the specified optimizer and loss function.

We defined the architecture of all the models in similar fashion so that it is easier to compare their results. While training, CNN for 7 epochs, biLSTM for 9 epochs GRU ran for 12 epochs. [5]

IV. Result & Analysis

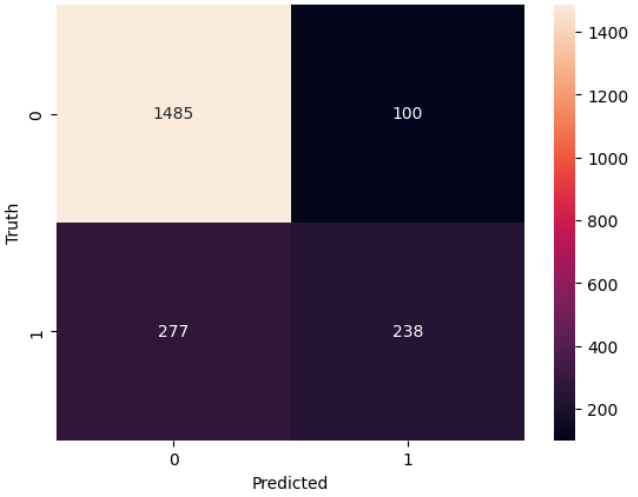
For analyzing the output of our models, we chose macro average f1 score and confusion matrix.

The F1 score is often used instead of accuracy when dealing with imbalanced datasets, where the number of samples in each class is not equal. Accuracy can be misleading in such cases, as it does not take into account the class distribution and may give an overly optimistic view of the model's performance. The F1 score is the harmonic mean of precision and recall, and gives an overall measure of a model's accuracy that is more informative than accuracy when dealing with imbalanced datasets. So, in our project we considered the macro average f1 score to determine the effectiveness of our model.

A confusion matrix is a table that summarizes the performance of a classification model by comparing the predicted labels to the true labels across different classes. For our binary classification problem, sexist texts were considered as 'positive' and not sexist texts were considered as 'negative'. A confusion matrix for this problem would have four entries: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). TP represents the number of sexist texts that were correctly classified as sexist, FP represents the number of non-sexist texts that were incorrectly classified as sexist, FN represents the number of sexist texts that were incorrectly classified as non-sexist, and TN represents the number of non-sexist texts that were correctly classified as non-sexist.

A. CNN Model

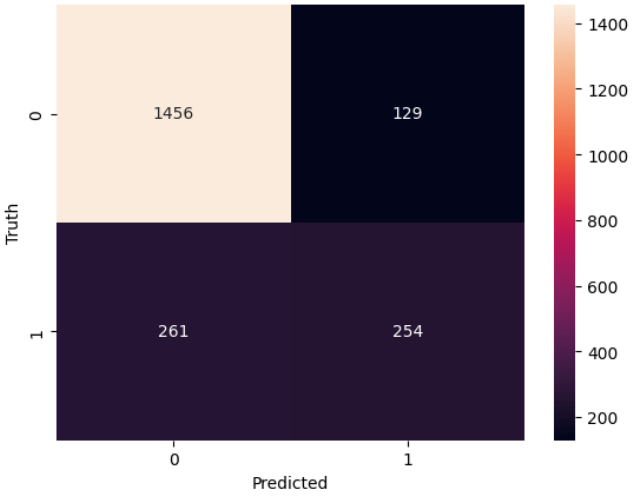
The macro average F1 score of CNN Model is 0.72 or 72%. Confusion matrix of CNN Model is



As the number of true negatives of our model is very high and it is an indicator of the effectiveness of the model. True positives were comparatively low due to imbalanced class.

B. Bidirectional LSTM

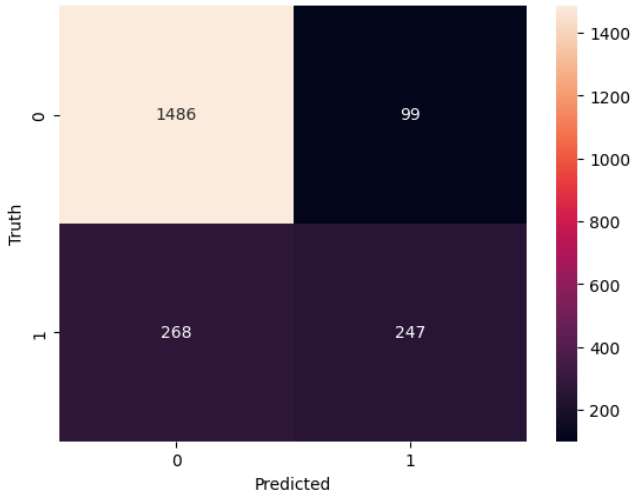
The macro average F1 score of biLSTM Model is 0.72 or 72% performing almost equally as CNN.



Here also we can see the number of true negatives are very high and close to CNN and true positives are comparatively low for data imbalance.

C. Gated Recurrent Unit (GRU)

The macro average F1 score of GRU Model is 0.73 or 73% performing little better than the other two models.



We can see similar true negative and true positive values here as well.

REFERENCES

[1] IBM. (2023) Convolutional neural networks. Accessed: May 10, 2023. [Online]. Available: https://www.ibm.com/topics/convolutional-neural-networks

[2] V. Zhou. (2019) An introduction to neural networks. Accessed: May 10, 2023. [Online]. Available: https://victorzhou.com/blog/intro-to-neural-networks

[3] e. a. Kirk, "Semeval-2023 task 10: Explainable detection of online sexism," 2023, accessed: May 10, 2023. [Online]. Available: https://arxiv.org/abs/2303.04222

[4] M. AI. (2019) The classification of text messages using lstm, bi-lstm, and gru. Accessed: May 10, 2023. [Online]. Available: https://medium.com/mlearning-ai/the-classification-of-text-messages-using-lstm-bi-lstm-and-gru-f79b207f90ad

[5] DatabaseCamp. (2023) Lstms. Accessed: May 10, 2023. [Online]. Available: https://databasecamp.de/en/ml/lstms