Detection of Sexually Harassing Tweets in Hindi using Deep Learning Methods

Shivaji Ray Chaudhuri  
Department of Computer Science and Engineering  
Manipal University JaipurJaipur, India  
shivajirchaudhuri@gmail.com

*Abstract*—In the modern era, social networking platforms play a vital role in day to day life. It is used by personalities, politicians, bureaucrats and common people all alike. It provides a platform for everyone to express their opinions, share their experiences with their family, friends and the world. The advent of such platforms has completely changed the outlook of the world.

But there are individuals or groups who use these social networking platforms to target people. Sexual Harassment has been a problem for a long time but with the advancement in technology, some users have taken the harassment to the digital level. It is common these days to find users posting derogatory remarks, uploading sensual and private content of others in order to sexually harass them. There are many reasons why people go down to such disgusting acts, they might be taking revenge from someone who has let them down or betrayed them in the past, some do it just to downgrade a person or groups in front of a society and some do it just for fun. But they often fail to realize that such acts might have disastrous effects on the targeted audience. They may feel humiliated, discriminated and might feel that something is wrong with them. This is not all, some targeted audience might even take some wrong steps.

This article presents a way to detect such derogatory and harassing remarks using Twitter as a database. A simple deep learning model will be used to do as such.

Keywords— Sexual Harassment, Sexual Harassment Detection, Deep Learning, LSTM, Twitter

# **Introduction**

Before we discuss about the more technical aspects of this article we need to know about the Social Networking Platform we are relying on.

Statistics have shown that as of November 2019 there are approximately 330 million monthly users in Twitter with approximately 145 million users using the service on a daily basis [21]. Since we are focusing on Hindi language we need to know the demographics of India as majority of the Hindi speaking crowd comes from India. Again as of 2019, India has approximately 7.75 million users on Twitter. Most people in India use Twitter as a source of news. There is very high gender inequality in the users as only 16% are female and rest 84% are male [4].

We also need to know what is considered as sexual harassment and its severity in social networking platforms. In general, sexual harassment refers to forceful or unwelcome gestures, behaviors or sexual advances directed towards a person. Mostly it is seen as a violence against women. But when it comes to sexual harassment on social networking platforms it takes a completely different perspective. Rather than advances as in physical harassment cases, in digital world sexual harassment is more related to humiliation, threatening or discriminating someone by posting, replying in a very unresentful, disturbing manner.

Talking about Twitter it was generally classified as a safe place for expressing views. The company even touted that every word has a power to change the world but these days sexually harassing tweets have become quite common targeting both men and women. Women have been targeted more with abusive, derogatory tweets. In a report from Amnesty International, they have described that when the world is taking huge strides in gender equality, some individuals or groups are using social medial platforms to curb the enthusiasm by posting offensive tweets which more often than not resulted in backing of women from such platforms and public fields [1].

This article focuses on Hindi tweets so we should also look about the situation of our country India which is the largest Hindi speaking nation. India has taken huge steps in promoting gender equality, prevention of discrimination, crime against women. These days’ women occupy top positions in the country’s governance and excel in all fields be it sports, business or social services. There is also a vast presence of presence of Indian women in Twitter. Again according to a report from Amnesty International where they surveyed 114,716 tweets for 95 of the top women politicians in India and found that 13.8% of the tweets were “problematic” or “abusive”, it contained harmful and hurtful content [3].

There have been many campaigns from welfare bodies around the world to spread awareness regarding the severity of such actions. They are directed towards men and women alike and needs to be controlled for a safe social media platform [2]. Thus this article presents a solution which classifies such tweets in Hindi language to be specific.

Now coming to sentiment analysis, this procedure allows us to classify texts into positive or negative classes. Since our research focuses on detecting sexually harassing tweets, our dataset contains tweets along with a binary polarity column that specifies whether a tweet can be classified as sexually harassing or not.

We have chosen a dataset such that majority of the tweets are in Hindi language. After that we have done some preprocessing on our data and then used a combination of CNN and LSTM model and a combination of RNN and LSTM model to obtain results. The technical part, accuracies achieved and conclusions will be discussed in the later parts of this article thoroughly

# **RELATED WORK**

## A lot of research has already been conducted on Sentiment Analysis. It is a very diverse field these days but still a lot of resarch is being carried out to increase the efficiences of machine so as to enable them to achieve human level efficiency. Most of the pre-existing works have focused their research on supervised learning techniques using common classifers like Support Vector Machine, K-Nearest Neighbors and Naïve Bayes algorithms [5] [6].

Work has also been done in this field with the use of conventional Deep Learning models like Recurrent Neural Networks and Convolutional Neural Networks to modify the approach. This approach did wonders as it has been proved that these neural networks often give better results than the existing methods where unigram and bigram models were used. These models do not only emphasize on the polarity of the statement but also heavily depends on the actual orientation of words in a sentence thus making the semantic aspect of the sentence as important as the other factors.

There are not many articles that clearly state that they are detecting sexually harassing [11]. There are articles that have done similar work like one of them used machine learning and sentiment prediction algorithms to detect suicide predictions in social networks. There are many articles that focuses on domain specific sentiment analysis. Domains like movie, politics and sports are the most common ones [10] [15].

Since this article is on detecting sexually harassing tweets in Hindi language it is important to know about the research that is being done or already has been done in the field of Sentiment analysis. Again excellent research has been done in these fields. In Hindi language, most of the research has been done based on a resource based lexical approach using HindiSentiWordNet (HSWN) [9] [12]. Even some unsupervised lexicon methods have been used. Again most of the research has been done using standard Machine Learning classifiers. There are also some articles who have demonstrated a deep learning approach for Hindi Language [7] [8]. Most of the research is often based on the datasets containing generic data, political data and movie data, these being the most common. IIT Patna has done some fantastic research in this field. They have focused on Aspect Based Sentiment Analysis in which they have used two techniques. One of them is Resource Creation and Evaluation in which they have used Conditional Random Field (CRF) and Support Vector Machine (SVM) classifiers and achieved an accuracy of 54.05% . The other method was Category Detection and Sentiment Analysis in which they have used Naïve Bayes and Decision Tree algorithms and classified various categories of data into positive and negative. In the same article they have also defined a CNN based approach to detect negative tweets and they have achieved an accuracy of 55% [13].

Most of the articles that have been studied as a part to initiate this project have been mentioned in the References. Most of these articles have been studied to understand their respective approaches, problems encountered, how problems were dealt with. Inspiration has been take from most of them while working on the project this article is based on.

# **PROPOSED METHODOLOGY AND NOVELTY OF IDEA**

Based on all the knowledge gathered, we decided to adopt Deep Learning models to train on our dataset. We have used two techniques, first is a combination of CNN and LSTM and second is a combination of RNN and LSTM. The technical details have been discussed in the later parts. Our choice of using Long Short Term Memory (LSTM) is simply because of its versatility when it comes to text related operations in machine learning. We want our model to capture the semantics of a sentence and make predictions based on it rather than to just rely on the numerical aspects only.

Before training the models, some basic preprocessing has been done. English alphabets, numbers, links, emoticons and new lines have been removed. Since any model cannot work on textual data, the tweets have been tokenized and later embedded in the model itself.

Now coming to the uniqueness of our idea, as it has already been discussed in the last section, Sentiment Analysis is a popular topic and lot of research has already been performed in it and so one of the most important question is, what new are we bringing to the table?

As briefed in our Introduction and Methodology, we are using Deep Learning methods to train our model on some data. The unique part is that we are fixing our domain and language to work on. There has been relatively fewer articles and research papers regarding the detection of sexually harassing texts. Since we are also restricting ourselves to Hindi language, thus it makes our research not only domain and language specific but also unique as no article has yet been published in this domain and Hindi language.

# **CHALLENGES**

Before initiating the project, we needed to address some of the challenges that were faced by others who had undertaken research in similar field and also the challenges that were imminent based on our decision to perform domain and language specific work. The challenges are discussed below

## Hindi Language

This was one of the first problems that we needed to consider as soon as we decided to fix our language to Hindi

### Different sentences can convey the same meaning: Hindi is spoken in various parts of the world and due to the regional diversity there are many variations in the language that come with it. Sentences can convey same meaning if only the word order is changed in some cases. Although one might debate that by changing the word order the semantic corectness of the sentence may be violated but we also need to consider that we are going to deal with live data from Twitter when predicting the sentiment so we need to prepare the model such as to encompass as much variations as we can.

An example showing the random word order problem in Hindi language is given below,

वह खाना खा रहा है

खाना खा रहा है वह

### Lack of tools to process Hindi texts: When it comes to process hindi text as compared to english text, it becomes a lot more challenging considering the scarcity of high end tools. A lot of research has already been done in English language. English language offers proper corpuses, stopword files, tools like stemmer, lemmatizer and a lot more processing techniques have been developed over the years.

But when it comes to Hindi texts there are some tools that are available but their effectiveness is still questionable. Corpuses have been developed for Hindi but they are scarce and often not relevant for some project requirements, stopword files are often custom made and not updated regularly.

## Dataset Problem

As it is already established that we are performing domain and language specific work, the problem that was inevitable was the lack of datasets firstly in Hindi and our domain of Sexual Harassment. Most of the Hindi datasets that were already available on the internet were not useful for our project as they contained very limited text that could be classified as sexually harassing and thus it would be not useful to train our machine on as there was a high probability that our model would not be able to identify any sexually harassing text later.

# **EXPERIMENT**

The implementation of the entire project is mentioned in a step by step manner in this section. This section will cover about the datasets, preprocessing and deep learning models in an elaborative manner.

1. Dataset Gathering – As it is already mentioned that we are using Twitter as a database and we are focusing on Hindi language.

We have used Twitter Intelligence Tool (TWINT) to fetch data from twitter. It is an advanced tweet scraping tool designed in Python that does not require us to provide any Twitter credentials while using it. This tool uses Twitter’s search operator to gather tweets based on keywords, from particular users, based on hashtags. It provides comprehensive functions that allow us to gather tweets in particular duration and also has a built in language support that filters tweets according to language [23].

Since there is no domain specific dataset available for us to properly train a deep learning model so we decided to create our own dataset and label them manually classifying them as either sexually harassing or not.

We started building our dataset by initially fetching tweets using TWINT and gathered around 6329 tweets for our training data and further around 2117 for our testing data. Now when building our dataset, we took notice of two things.

Firstly, we needed to train our deep learning model as accurately as possible, and we noticed that during the normal fetching of tweet there were a very limited number of tweets that can be classified as sexually harassing. For that we used the TWINT function that allows us to search tweets by keywords. The keywords were decided after going through some websites that listed words that could be classified as sexually harassing. The keywords will not be disclosed in this section but a link to it will be present in the references. The keywords were purely chosen so as to train the model as accurately as possible.

Secondly to ensure that the machine has been trained properly we collected the training and testing data from different time periods. The training data was fetched from recent tweets whereas the testing data was fetched from the time period of 2013-2015. By doing this we wanted to ensure that our machine trains in such a way that it rather captures the semantic aspect of sentences and does not get trained on a particularly popular topic that is being tweeted in that period. Thus by doing this we ensured that our training and testing data are completely containing different mindset of thoughts and expressions and our machine can only detect the tweets that are sexually harassing.

|  |  |  |
| --- | --- | --- |
|  | **Training Data** | **Testing Data** |
| **Positive** | 2827 | 1016 |
| **Negative** | 3502 | 1101 |

1. Preprocessing of Data – The preprocessing of data mainly consisted of five major steps.

* The first step was to replace all the new line characters from the tweet with a space. This step significantly helped in increasing the efficiency. It also resulted in a more compact dataset.
* The second step was to remove all emoticons from tweets. It is a popular method to express views these days but considering each one of them would significantly increase the load on the model. A function was defined to remove the emoticons using their Unicode representation.
* The third step was to remove all the other characters that decrease the effectiveness of model. They include numbers, special characters and also English alphabets. The necessity for removing English alphabets was simply to allow the model to focus purely on Hindi characters and words.
* The fourth step was important in a machine learning perspective as we performed tokenization and vectorization of the textual data that is the tweets. We utilized the Tokenizer class from Keras library of python to tokenize the data. It converts the words of a sentence into sequences and also performs vectorization according to the most common words. The most common words are specified by the user and in our case it was 1000. It generally encodes a word by its occurrence. Higher the frequency, lower will be its value signifying a higher rank.
* The final step in the preprocessing part is to insert padding either at the end or beginning. To do that we also need to mention maximum length of allowed texts. We know that most of the tweets are not of uniform length and vary but we cannot pass varying length sequences to deep learning algorithm so we need to specify a maximum length. If the length of sequence is greater than the sequences will be cut short either from the beginning or the end. If the length of sequence is short, then it gets padded with a value at the end or start which is usually pre-defined. Thus this steps ensure that our data is ready to be fed to the deep learning models.

1. Model used and Description – As we have already established that we are going to use Deep Learning models in this article, thus in this section we are going to give a detailed explanation about the model used, the input and output shape and other necessary details. We have accomplished this project with the help of two models, both of them being Sequential models. Before we describe the models used in this project we would like to elaborate on some key terms that are beneficial for understanding the upcoming parts.

* Sequential Model – Since we are using most of the features from Keras library, we need to know about the sequential model that is being used in both models in this project. It is one of the simplest ways to create a neural network. As the name suggests we add necessary layers in a sequential manner in this type of model. This type of model allows the data flow in a linear fashion that is the output of one layer will be the input of another layer.
* Dense Layer – Again it is one of the most used layers in Keras Library. It is called Dense layer because all the incoming nodes are connected to all the output nodes from that layer. It is mostly used to change the shape of vectors that are being passed onto it. The output of a dense layer is usually calculated at each node by the formula mentioned below: -

Output = a(dot\_product(input, k) + b)

Here ‘a’ is the activation function, ‘k’ is a weight function generated by the layer itself and ‘b’ is the bias vector which is generally used to delay the activation function unlike weight vector which decides how fast the activation function will be activated.

* Activation Function – They are a fundamental part of any deep learning model as this is one of the aspects that differentiate machine learning models from deep learning models. Its basic function is to convert input signal to output signal. A deep learning model without any activation function behaves as simple linear regression model and thus to add the additional complexities and dimensions we use activation functions [19]. Two activation functions used in this article.

The first one of them is probably the most common one. It is the sigmoid activation function which returns a value between 0 and 1. It is calculated by: -

**f(x) = 1 / 1 + exp(-x)**

The second activation function used is Rectified Linear Units (relu). It is a very simple activation function which is mostly used in hidden layers of neural networks. It is calculated by: -

**R(x) = max(0,x) i.e. if x < 0 , R(x) = 0 and if x >= 0 , R(x) = x.**

* Dropout - When using machine learning and deep learning models, one of our primary goals is to fit the model as accurately as possible but in doing so we also need to ensure that we do not overfit our model and this is where dropout comes into play. Usually dropout is assigned a value between 0 and 1 and then it drops the corresponding percentage of nodes when training.
* Embedding Layer in Keras – It is a known fact that just by tokenizing our dataset we cannot achieve good accuracy. For our requirements we needed something that can embed words in such a way that similar words have similar vector values or that similar words will be close to each other in a multi-dimensional space so that close relations and associations can be recognized between words.

The three necessary parameters in this layer are described ahead. First is the **input\_dim** which specifies the vocabulary size we are opting for. Next is **output\_dim** which specifies in how many dimensions a word will be embedded into. Last is the **input\_length** which specifies the size of sequence or the length of sentence. It is common for all sentences and this is where the padding of sentences comes into play.

* CNN – It stands for Concurrent Neural Networks which is part of a deed forward and deep neural networks. It is named so, because of the hidden layers it encompasses. Hidden layers like convolutional, pooling and normalization layers are common and the activation function is performed in these layers to obtain result. It is mostly used in computer vision but recently it has found its use in text classification also. In Keras we are using the Conv1D function which creates a single dimensional convolutional network.
* RNN – It stands for Recurrent Neural Networks. As the name suggests this type of neural network consists of nodes which form blocks that are connected to each other in a sequence. It forms a directed graph between nodes along a sequence. If used with proper embedding it is one of the best methods to perform text classification. In Keras we have used the LSTM function to utilize the properties of a recurrent neural network.

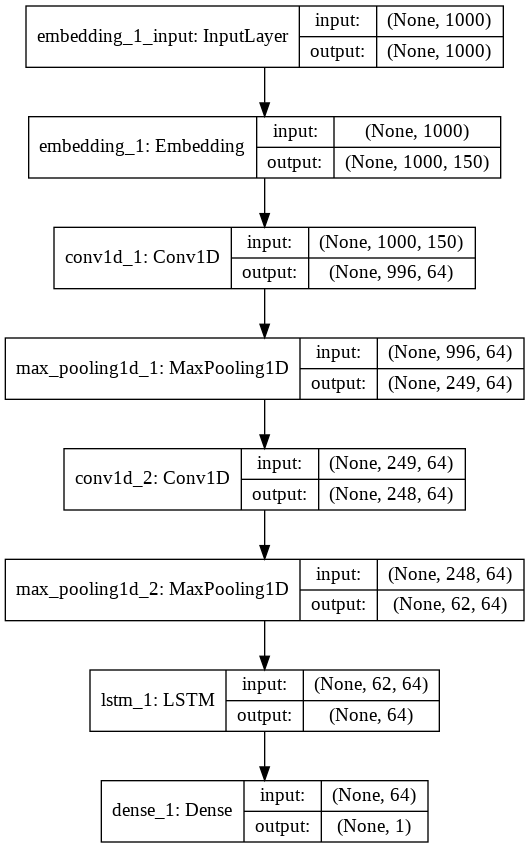
LSTM stands for Long Short Term Memory and is the most useful feature for our article. We have used LSTM in both of our models which will be soon discussed. Since recurrent neural network only has a short term memory that is it relies on the information that has been learned recently. LSTM introduces a long term memory and hence it is very practical for semantic analysis of sentences as it can easily learn from a lot of data and make predictions on the basis of what it has learned from the whole dataset rather than a small subset of it [16] [18].

* MaxPooling1D in Keras – MaxPooling1D is a layer in Keras which is used for pooling operations.

Pooling is used to reduce the overall complexity of model. It is mostly used in unison with convolutional layers in a CNN for allowing it to minimize distortions. But the main function of Pooling layer is to reduce the dimensions of data from the previous step and in doing so it does not affect the training time either as it does not require any additional parameters.

Now since we have familiarized ourselves with the terms that we are going to come across, we will now describe the model used.

* The first model that we have used to classify sexually harassing comments is the CNN-LSTM model. A representation of the various layers used is given below.



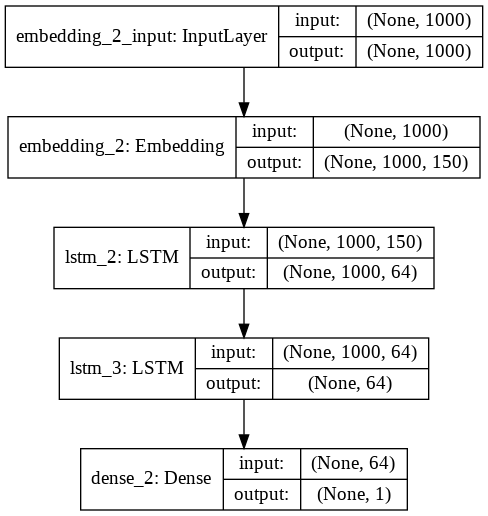
As one can observe in the model the first layer used is the embedding layer in which each word is represented using a 1000-dimensional vector as one can see in the output shape.

Next is a CNN layer which consists of two consecutive blocks of convolutional and max\_pooling1D layers. Their purpose has been described above.

Next is a LSTM layer which is the most important part of this model and finally there is a dense layer which produces the required outputs.

Now we discuss the idea behind using a combination of two distinct neural networks. Both of them have different approaches when dealing with text classification and thus an attempt was made to capture the benefits of both these models. As we have seen initially the data is embedded then fed to the CNN model, the purpose behind this was to extract a sequence of higher level semantic representations which are then passed onto the LSTM model to extract the sentence representations, the benefit of this is to not only train the machine according to the semantics but also focus on the local features of phrases in that sentence [20].

* The second model used in our project is the more traditional RNN-LSTM model. A representation of the various layers used is given below.



As seen from the diagram above, this is a smaller model as compared to the last one. This also consists of an embedding layer.

Following the embedding layers are two LSTM layer. The important thing to note here is that the first LSTM model downscales the dimensions of the input vector matrix and the second LSTM model further downscales it.

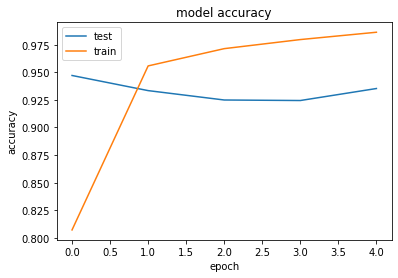
Finally, there is a dense layer which has a single output node to finally generate the required results.

Now coming to the intuition behind going for such an approach. The above model is a stacked LSTM model evident from the fact that the hidden layer output from the outer LSTM model acts as an input for the inner LSTM model. One of the main reasons for adopting such a stacked model is to increase the depth of the hidden layers. It is often said that deeper the neural network more accurately it can predict. Such approaches have been developed for a lot of text classification techniques. Coming to the benefit of having deeper neural networks, what basically happens in the background is that the hidden inner layers receive representations from the outer layers and they create new representations for these features at higher levels of abstraction. Also it is noteworthy that just by adding two layers of LSTM model we can achieve a very high accuracy for our predictions and thus reduces the representational complexity. Multiple LSTM layers are very useful for sequence prediction problems and thus this is what we intend to use in our project as we want to capture the order of words in a sentence that affects the polarity or it sentiment [22].

# **Result and observations**

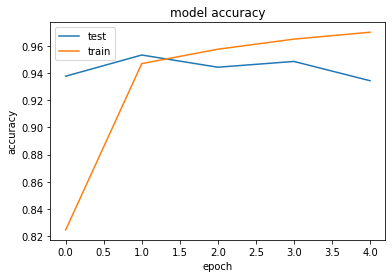
Now since we have gone through the models layer by layer. Thus in this section we will go through the results, the accuracies achieved by the models and their statistical comparison.

Coming to the first model that is the CNN-LSTM model, it resulted to an accuracy of **93.53%**. The model was trained over 5 epochs and the accuracy plot between the training and test data is plotted below.



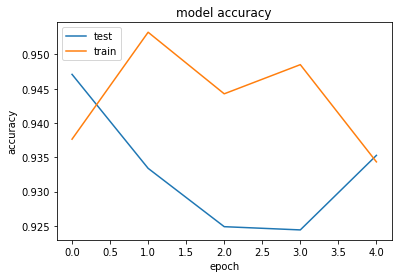
From this plot we can observe that there is a steep increase in the training accuracy over the first epoch and after that there is a gradual increase, but we are more concerned with the test data accuracy which is stable over the epochs depicting a perfect fit and excellent model performance.

Next is the RNN-LSTM model or the Stacked LSTM model, it resulted in an accuracy of **93.43%**. This model was also trained over 5 epochs and the accuracy plot between the training and testing data is plotted below.

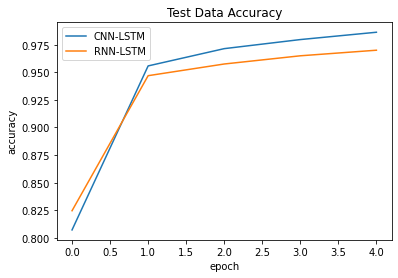


A slight contrasting plot than before, the testing accuracy plot shows two local maxima’s which might be as a result of some unexpected distortions although they are very minimal in terms of value. This model also results in a perfect fit and excellent model performance.

Now we need to compare both the models on the basis of performance. A few statistical plots and their observations are given below.



In the following plot the blue line denotes testing accuracy for CNN-LSTM and green line denotes testing accuracy for RNN-LSTM. Both of the have quite contrasting plots. We can observe that CNN-LSTM has a high accuracy at start but decreases until 3rd epoch after which it increases. Comparing to that the accuracy of RNN-LSTM varies quite regularly with the epochs and after the 3rd epoch it decreases. At the end both of them almost come to the same conclusion.



A more proper plot than the testing accuracy. The RNN-LSTM initially starts slightly higher than the CNN-LSTM but after the first epoch CNN-LSTM dominates. This might be as a result of those convolutional layers along with the pooling layers which eliminates distortions from previous epochs.

Now we take a look at the confusion matrix of both the models.Starting with the CNN-LSTM model.

|  |  |
| --- | --- |
| 981 | 35 |
| 102 | 999 |

From this we can observe that the CNN-LSTM model has done a very good performance in classifying the sexually harassing tweets and the non harassing tweets. Also there are 35 Type-1 errors and 102 Type-2 errors.

Now we take a look at the confusion matrix of RNN-LSTM model.

|  |  |
| --- | --- |
| 905 | 111 |
| 28 | 1073 |

The RNN-LSTM model also has performed a very well classfying the tweets. This model results in 111 Type-1 errors and 28 Type-2 errros.

On comparing the models, both have performed exceptionally well considering they are processing Hindi language. CNN-LSTM slightly edges out the RNN-LSTM model by **0.1%.**

But when it comes to business point of view the RNN-LSTM model slightly outweighs CNN-LSTM model. This observation is made considering the confusion matrix. Though both the models have almost same number of correct predictions but the number of Type-2 errors are more in CNN-LSTM which means that it has classified sexually harassing tweets as not harassing tweets which is a bit of a problem as our aim is to identify as many sexually harassing tweets as possible. So in this situation, RNN-LSTM has lesser number of Type-2 errors.

# **CONCLUSION**

In this article we discussed about detecting sexually harassing tweets using two deep learning models and thus compare their statistics and behavior. For doing so we manually created a domain specific dataset, performed preprocessing and trained it using the models. The models used are CNN-LSTM and RNN-LSTM (also known as Stacked-LSTM). Both of the models have successfully captured the semantic aspects and important features of the tweets. The performance of both the models are much better than the existing machine learning and standard deep learning models. In conclusion, we can say that we have been successful in detecting sexually harassing comments using the above mentioned deep learning approaches.

# References

1. <https://www.amnesty.org/en/latest/research/2018/03/online-violence-against-women-chapter-1/>
2. <https://www.amnesty.org/en/latest/research/2018/03/online-violence-against-women-chapter-8/>
3. <https://amnesty.org.in/news-update/shocking-scale-of-abuse-on-twitter-against-women-politicians-in-india/>
4. [https://www.talkwalker.com/blog/social-media-statistics-in-india#](https://www.talkwalker.com/blog/social-media-statistics-in-india)
5. Sentiment Analysis of Twitter Data

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau. [Link](https://www.aclweb.org/anthology/W11-0705.pdf)

1. Semantic Sentiment Analysis of Twitter by Hassan Saif, Yulan He, Harith Alani. [Link](https://link.springer.com/chapter/10.1007/978-3-642-35176-1_32)
2. Twitter brandsentiment analysis**:** A hybrid system using n-gramanalysis and dynamicartificial neural network by M Ghiassi, J Skinner, D Zimbra. [Link](https://www.sciencedirect.com/science/article/abs/pii/S0957417413003552)
3. A combined CNN and LSTM Model for Arabic Sentiment Analysis by Abdulaziz M.Alayba, Vasile Palade, Matthew England, Rahat Iqbal. [Link](https://link.springer.com/chapter/10.1007/978-3-319-99740-7_12)
4. A Framework for Sentiment Analysis in Hindi using HSWN by Pooja Pandey, Sharvari Govilkar. [Link](https://www.semanticscholar.org/paper/A-Framework-for-Sentiment-Analysis-in-Hindi-using-Pandey-Govilkar/e2a6a01802188ed32f72fd556a5d7c308e03e4e4)
5. Prediction of Indian Election using sentiment analysis on Hindi Twitter by Parul Sharma, Teng-Sheng Moh. [Link](https://ieeexplore.ieee.org/abstract/document/7840818)
6. A practical approach to Sentiment Analysis of hindi tweets by Yakshi Sharma, Veenu Mangat, Mandeep Kaur. [Link](https://ieeexplore.ieee.org/abstract/document/7375207)
7. A Fall-back Strategy for Sentiment Analysis in Hindi: a Case Study by Aditya Joshi, Balamurali A R, Pushpak Bhattacharya. [Link](https://www.researchgate.net/profile/Aditya_Joshi9/publication/280998440_A_Fall-back_Strategy_for_Sentiment_Analysis_in_Hindi_a_Case_Study/links/55d01dc208aee19936fd9cbe.pdf)
8. Sentiment Analysis by Md Shad Akhtar, Department of CSE, IIT Patna. [Link](http://www.iitp.ac.in/~shad.pcs15/data/sa-shad.pdf)
9. Machine Learning and Semantic Sentiment Analysis based Algorithms for Suicide Sentiment Prediction in Social Networks by Marouane Birjali, Abderrahim Beni-Hssane, Mohammed Erritali. [Link](https://www.sciencedirect.com/science/article/pii/S187705091731699X)
10. Domain-specific sentiment analysis using contextual feature generation by Yoonjung Choi, Youngho Kim, Sung-Hyon Myaeng. [Link](https://dl.acm.org/doi/abs/10.1145/1651461.1651469)
11. <https://medium.com/datathings/the-magic-of-lstm-neural-networks-6775e8b540cd>
12. <https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f>
13. <https://hub.packtpub.com/what-is-lstm/>
14. <https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f>
15. <https://arxiv.org/abs/1511.08630>
16. <https://www.oberlo.in/blog/twitter-statistics>
17. <https://machinelearningmastery.com/stacked-long-short-term-memory-networks/>
18. <https://github.com/twintproject/twint>