REVIEW 3

Cyberbullying detection using machine learning techniques

Offensive Meme Detection

CSE3502

Information Security Management

Slot: F1

in

B.Tech (IT)

by

Piyush Sahu (19BIT0038)

Varun Kumar (19BIT0058)

Amartya Sharma (19BIT0021)

6th semester, 2022

Under the Guidance of

SUMAIYA THASEEN I

SITE

Google drive link:https://drive.google.com/drive/folders/1wUKtpjgYZsU3aKOpvXeg_Z6-zBQHkpZI?usp=sharing

1.Abstract

A meme is a type of digital media that transmits an idea or feeling. The detection of hate speech, offensive content, and aggression content in a single model, such as text or image, has been intensively investigated. Combining two models to detect offensive content is something difficult. Memes make it even more difficult because they implicitly express comedy and sarcasm, thus the meme may not be offensive if we merely look at the text or image but the combination of both can result in it being offensive.

As a result, combining graphics and words to determine whether a meme is inappropriate is required. Thus first a model is created which will work on text only and then an image model is created then together the text model and image model is combined resulting in offensive meme detection. The dataset was obtained from kaggle, which had an image, its text that is written on it and some useless columns that were dropped and some preprocessing was performed. Then pretrained models are used like GloVe for tokenization of text which is passed to models like CNN, naive Bayes, LSTM, BiLSTM etc.and for image VGG16 was used. Then we will combine the base text model and image model together to detect offensive memes.

2.Introduction

With the growing popularity of social media these days , meme - a type of digital media which can be described as an element of a culture or system of behavior passed from one individual to another by imitating or other non-genetic behaviors has been widely used in all these social media platforms . It comes with various formats i.e. memes in forms of images , videos etc. . In this project we have decided to work on the most common type of meme used these days which is memes as images containing text in them .

Unfortunately, these memes which are supposed to be informative and enjoyable are being used for spreading hatred in our society . With this project of ours we focus on detecting such memes so that any offensive content which does not belong to the social media platform can be immediately reported and necessary actions can be taken regarding this issue . Memes are classified as offensive/abusive if it consists of racial, homophobic, or other offending slurs.

As a meme consists of both image and captions(in form of text) it becomes quite difficult to classify a meme as offensive or non - offensive just by looking at one parameter at a time. Thus we have come up with such a model which considers both the textual data as well as the image for predicting the output class. Machine learning models such as logistic regression , Naive Bayes and deep neural networks were used for training the textual data initially and VGG16 was preferred for working on the image data . Finally a combined model which takes in BiLSTM , CNN , Stacked LSTM with VGG16 was used to work on the overall meme dataset .

3.Literature survey

Amartya Sharma - 19BIT0021

[9] In this paper the authors tried to detect hate speech with the concept of transfer learning . The hateful texts taken to perform the research belonged to tweets from twitter , in total 37,520 tweets were used which consisted of both harmless messages and racist/sexist comments. The authors used deep neural network architecture with transfer learning (t deep hate) along with this proposed methodology , they introduced - Map of Hate a 2D visualization of racist/sexist content, which is helpful in differentiating types of hateful content. Their architecture is initially based on Bi directional LSTM and then using the concept of transfer learning for hate speech prediction . Their method predicts with the micro – averaged F1 score for 78% and 72% % in the first and second task, respectively.

[10] The authors proposed an automated system based on CNN to detect hateful commen ts on social media platforms . They also used a pre-trained embedding vector GloVe which works by converting textual comments to numerical vectors. The proposed system was also found better as it was able to perform accurately even with an imbalanced dataset . They also performed experimentation with the existing models like LSTM and Convolutional LSTM but the proposed model outperformed these existing models . The results achieved as 0.97 , 0.88 and 0.92 for precision , recall and F1 score respectively.

[11] In this paper the authors have experimented on various hate speech datasets which are Wassem and Hovey , SemEval and Covid – Hate(collection of hate speech during US elections related to covid 19). They proposed A – stacking classifier which is a hybrid classifier based on ensemble learning . The authors made sure this proposed model is adaptive in nature as a result of A stacking classifier. Also Recurrent neural networks were used in set up for generating word embeddings of tweets, this embedding layer was followed with LSTM network . The results of the proposed model on Waseem and Hovy dataset was promising as it achieved 81.54 F1 score, 82.21 precision and 81.38 as recall.

[12]The authors through this paper aims to prevent cyberbullying by harmful content which are in format of textual ,visual and info graphic. CapsNet-ConvNet , the architecture proposed by them is a combination of CapsNet a deep neural network with ConvNEt a convolution neural network. They used over ten thousand hate speech content from various social media platforms . The AUC-ROC score devised from this experiment was 0.98 and also a comparison of this proposed model with the conventional algorithms such as KNN, SVM and Naïve Bayesian was performed , which resulted in success of proposed model over the ones.

Piyush Sahu - 19BIT0038

[5] In this author aimed to detect hate speech like racism, sexism, homophobia, xenophobia etc. in tweets from twitter using deep learning techniques. First a baseline is created using char-n-gram ,TF-IDF and BoWV with embedding and then deep learning models are created consisting of GloVe or random embedding on CNN, FastText and LSTM. Also another experiment is done consisting of these methods and GBDT classifiers. The best performing model was LSTM+Random embedding+GBDT which had Precision , recall and F1 as 0.93, 0.93, 0.93. In base paper this paper showed that racism is mostly on negative posts and DLL is 18% more effective as compared to conventional methods.

[6] Here the author has done sentiment analysis on tweets regarding the Syrian refugee crisis in English and Turkish language. A sentiment dictionary was created for Turkish language and for English a pre-existing dictionary was used which classified tweets from very negative to very positive using a score. It was found that Turkish tweets (35%) were more positive than English tweets(12%). Also, Turkish and English tweets were different in terms of what was being discussed. This paper helped by showing that the higher accuracy, precision and recall, the better the model.

[7] In this paper the author has aimed to detect fake news in online space using 23 supervised AI algorithms. Here data is converted to unstructured to structured data by storing in vector using TF weighting and document term matrix. Thus, twenty-three AI algorithms are applied on it and later compared using 4 evaluation metrics. Decision tree had the best mean accuracy, F-score and precision (0.745, 0.741,0.759)whereas CVPS had best recall(1.000). This paper helped base paper by showing that number of followers is also indicator in fake news detection and also acted as proof that this type of study is being conducted.

[8] In this paper the author used machine learning and sentiment analysis to detect cyberbullying in online in-game chats. Firstly, data was collected for in-game chat and combined with user's information. The data was classified using SQL classifier, sentiment analyser and custom-built classification client. It was found that experienced people do more cyberbullying and usually happens just after the player is killed. This paper was used in base paper as a poof that studies have been done regarding cyberbullying using ML.

Varun Kumar - 19BIT0058

[1] In this paper the authors have proposed a new approach involving a deep learning model called HCovBi-Caps for hate speech detection. This approach involves integration of convolution layer, BiGRU layer and the capsule network lawyers for detection of hate speech. The capsule network helps incorporate contextual information at different orientations. The proposed model has been evaluated over 2 - Twitter based benchmark datasets of which one is balanced and the other is unbalanced. The model outperformed other state-of-the-art and baseline methods with precision, recall, f - score values of 0.90, 0.80 and 0.84 respectively. Some limitations of HCovBi-Caps model is that it detects hate speech in text only and does not exploit the users' profile related features.

[2] In this paper, the authors have proposed a novel GCR-NN model for the purpose of detecting tweets that contain racist content. The GCR-NN model is an ensemble of three deep learning models - Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), and Recurrent Neural Network(RNN) - stacked on top of each other. GRU extracts the prominent features from raw text, CNN extracts several other features which are used by RNN to make accurate predictions. The dataset was collected from twitter and annotated using TextBlob. The proposed GCR-NN outperformed the existing models by correctly detecting 97% of the racist tweets with a misclassification rate of only 3%.

[3] In this paper, the authors have proposed a novel deep learning model called Multi-View Attention Network (MVAN) for detecting fake news on social media networks. The MVAN combines two attention mechanisms, text semantic attention and propagation structure attention. Hence, MVAN is capable of capturing important information in the source text and the propagation structure of the tweet. MVAN has also been used for classification tasks such as sentiment classification, insult detection, etc. The evaluation results show that MVAN outperforms other baseline methods by 2.5% in average accuracy.

[4] In this paper, the authors have proposed a novel approach for detecting hate comments online using Reinforcement learning. The proposed model, Q-Bully, uses human-like behavioural patterns combined with Natural Language Processing techniques to detect cyberbullying. Delayed reward concept has been used so that the model considers the overall sentence and not just a part of it. NLP has been used to amplify the results by improving the convergence rate in the event of a high rate of exploitation. The F1 score of the proposed model turned out to be 0.86. The model clearly outperformed the baseline methods when the dataset is dynamic and consists of relatively newer words.

Tabular Column

	Accuracy	Precision	Recall	F-score
[1]	-	0.90	0.80	0.84
[5]	-	0.93	0.93	0.93
[7]	0.745	0.759	-	0.741
[10]	-	0.97	0.88	0.92
[11]	-	.8221	.8138	.8154

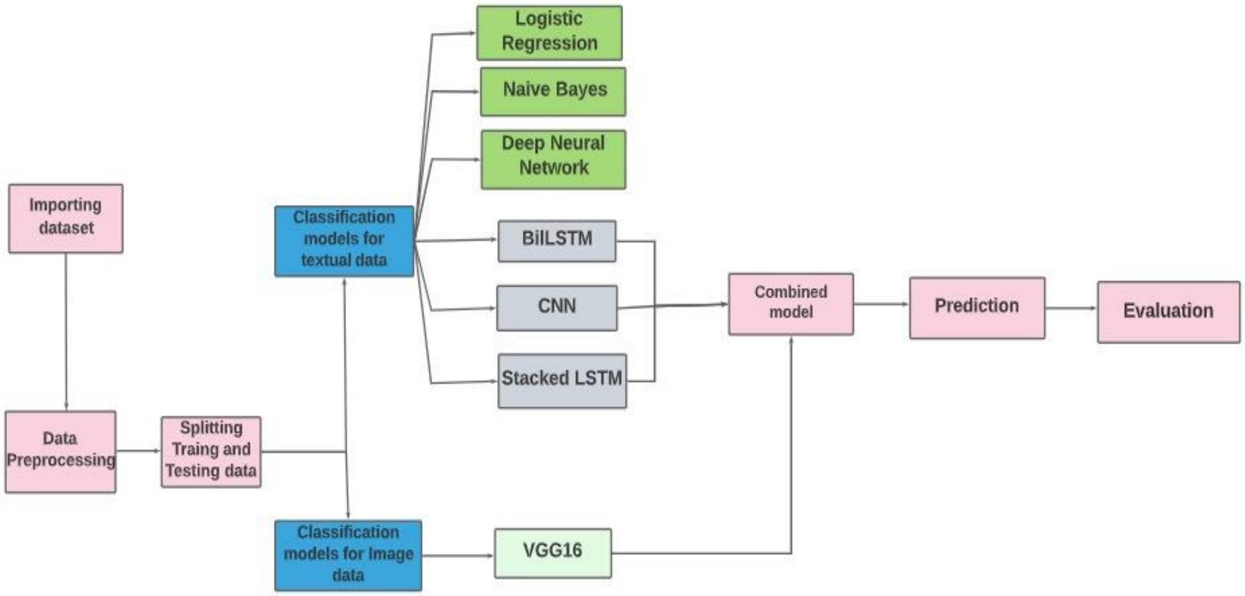
(-) Means given metric not mentioned in the paper

4. Proposed methodology

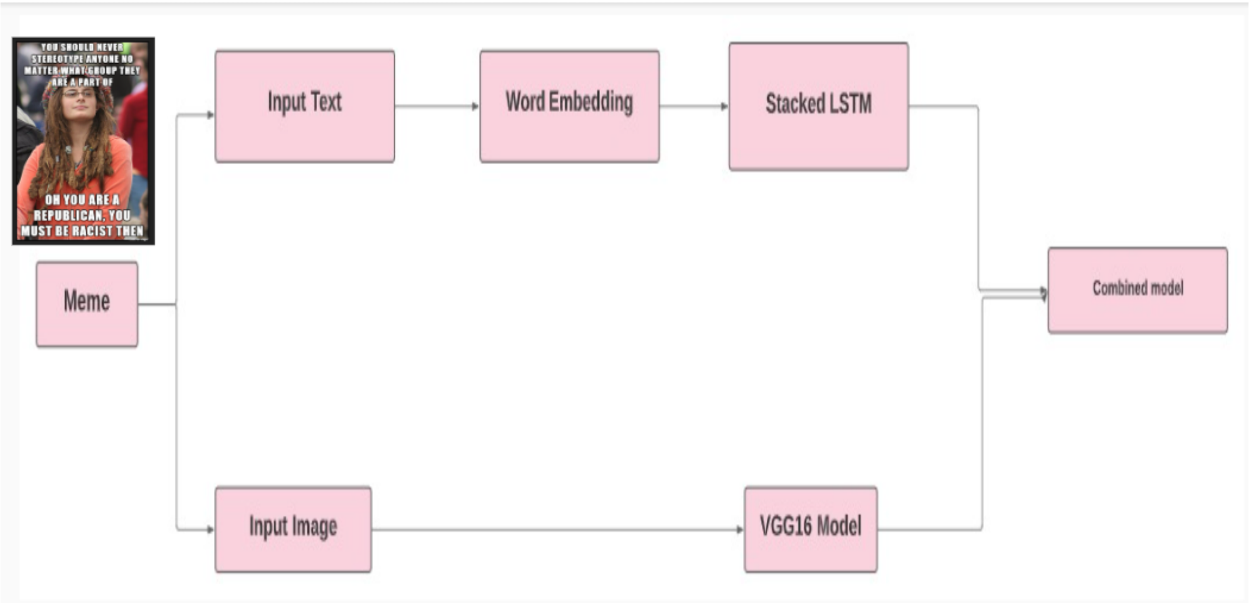
To detect offensive meme, we decide to split the meme into two part - text and image. When both are individually seen and read, they may or may not appear offensive but when the text is given a context with respect to image then the meme may become offensive.

Thus first we have analysed the text independently using model like logistic regression, naive bayes, Neural networks, CNN, BiLSTM and Stacked LSTM. These model were trained for textual data and were saved for later purpose. We used VGG16 model to train for image. Finally three combined models were proposed which were a combination of – Stacked LSTM(For text) + VGG16(Pre-trained for image) , BiLSTM(For text) + VGG16(Pre-trained for image) and CNN(For text) + VGG16(Pre-trained for image) .Then performing a comparative analysis between the proposed models and selecting which one best detects and classifies memes into offensive and non-offensive category.

High Level Diagram



Low Level Diagram


YOU SHOULD NEVER STEREOTYPE ANYONE NO MATTER WHAT GROUP THEY ARE A PART OF
OH YOU ARE A REPUBLICAN, YOU MUST BE RACIST THEN
Meme
Input Text
Word Embedding
Stacked LSTM
Combined model
Input Image
VGG16 Model

Modules and its description

Importing of and Dataset Data Preprocessing

This dataset on which our machine learning models are trained is collected by means of web scraping various websites and social media platforms . As the data we are handling data which consists of text as well as image we first need to pre - process this data to make it fit for further use in model building . The pre-processing includes some crucial steps such as removing stop words(unused words) , bad words and also the use of label encoders to convert label class of offensive and non -offensive to binary format of 0’s and 1’s . Then we do some preprocessing on the image. Some preprocessing techniques used include Pixel brightness transformations/ Brightness corrections, Geometric Transformations, Image Filtering and Segmentation, Fourier transform and Image restoration. We also perform image augmentation to increase the dataset size.

Classification Models for textual data

Logistic Regression , Naive Bayes Deep & Neural Networks

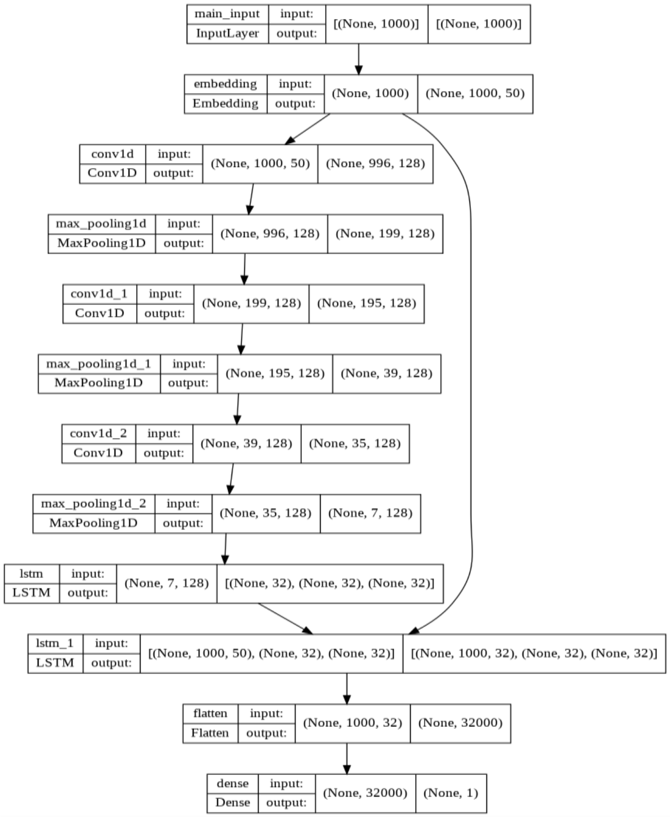
Initially we trained and tested our dataset with the three machine learning models namely - Logistic regression , Naive bayes and Deep Neural Networks These models are trained on textual data only , this is to know how accurately these models classify the captions present in a meme and define them as offensive or non - offensive . The inspiration of using logistic regression and Naive bayes models came from the base paper and use of Deep neural Network was to compare how well it performed in comparison to thesis models . The following process used in this module have been summarized below :

- Storing the textual data in an array data structure
- Converting textual data to numerical form with the help of python libraries such as scikit learn and keras .
- Model training (Logistic Regression , Naive Bayes)
- Model Building (DNN)
- Output(Precision , Recall , F-score , Support)

Convolutional Neural Network

Here a CNN model is created which consists of three 1D-convolution layers with 3 max pooling layer, 2 LSTM layer , 1 input, embedding, flatten and dense layer. This model is used to classify offensive text of memes as offensive and non offensive, thus a lot of preprocessing of data is done on text. Preprocessing includes removing unwanted characters, spaces, emojis etc. Then the pretrained GLoVe model is loaded for word vectorization which helps to create an embedding layer of CNN. Then the layers are created and training is done, and finally a confusion matrix is created.

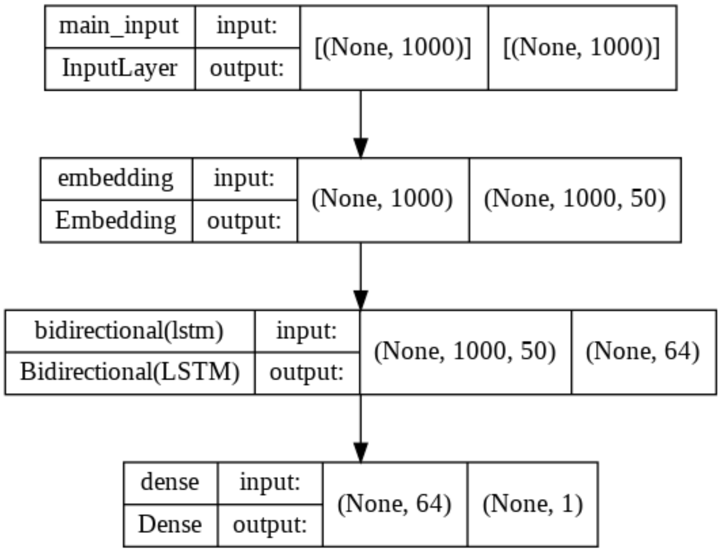
Architecture of CNN model that is being implemented


main_input | input: | [(None, 1000)] | [(None, 1000)]
InputLayer | output:
embedding | input: | (None, 1000) | (None, 1000, 50)
Embedding | output:
conv1d | input: | (None, 1000, 50) | (None, 996, 128)
Conv1D | output:
max_pooling1d | input: | (None, 996, 128) | (None, 199, 128)
MaxPooling1D | output:
conv1d_1 | input: | (None, 199, 128) | (None, 195, 128)
Conv1D | output:
max_pooling1d_1 | input: | (None, 195, 128) | (None, 39, 128)
MaxPooling1D | output:
conv1d_2 | input: | (None, 39, 128) | (None, 35, 128)
Conv1D | output:
max_pooling1d_2 | input: | (None, 35, 128) | (None, 7, 128)
MaxPooling1D | output:
lstm | input: | (None, 7, 128) | [(None, 32), (None, 32), (None, 32)]
LSTM | output:
lstm_1 | input: | [(None, 1000, 50), (None, 32), (None, 32)] | [(None, 1000, 32), (None, 32), (None, 32)]
LSTM | output:
flatten | input: | (None, 1000, 32) | (None, 32000)
Flatten | output:
dense | input: | (None, 32000) | (None, 1)
Dense | output:

Bidirectional LSTM

As recurrent neural networks work best on sequential data whether it be sequential data in form of video , image or text . Bi LSTM or Bi directional Recurrent Neural Networks are does not only take in consideration of the present and the past inputs but also take in consideration of the future events as it happens sometimes in a textual data some text only makes sense when it is fully complete. A bi directional recurrent neural network is a combination of two RNN’s - one moves forward , beginning from the start of data sequence and other moves backward, beginning from end of data sequence.

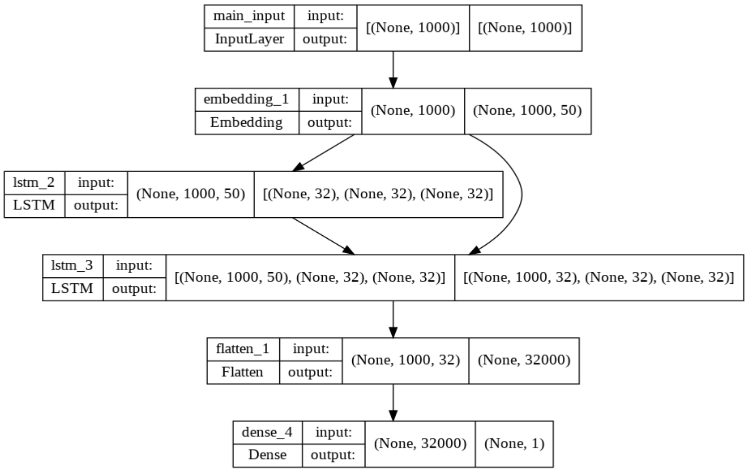
In this project we have applied Bi LSTM with three layers namely an input layer with max_len as one of the parameters , an embedding layer and a bidirectional LSTM sequential layer with 64 output dimensions and finally with a dense layer which is the output layer with one as output dimension.



Stacked LSTM

Other approaches don't preserve the context of the word but with LSTM, the data is treated as a time series data thus important words are remembered and a context is derived without the problem of vanishing gradient descent. Stacking the LSTM model allows greater complexity which helps to describe complex patterns in sentences.

In our project we stacked two LSTM layers of 32 input size each, they are preceded by an input layer and embedding layer and proceeded by flattening layer and dense layer for output. The model is using GLoVe word embedding for training and testing.



Classification Model for Image data

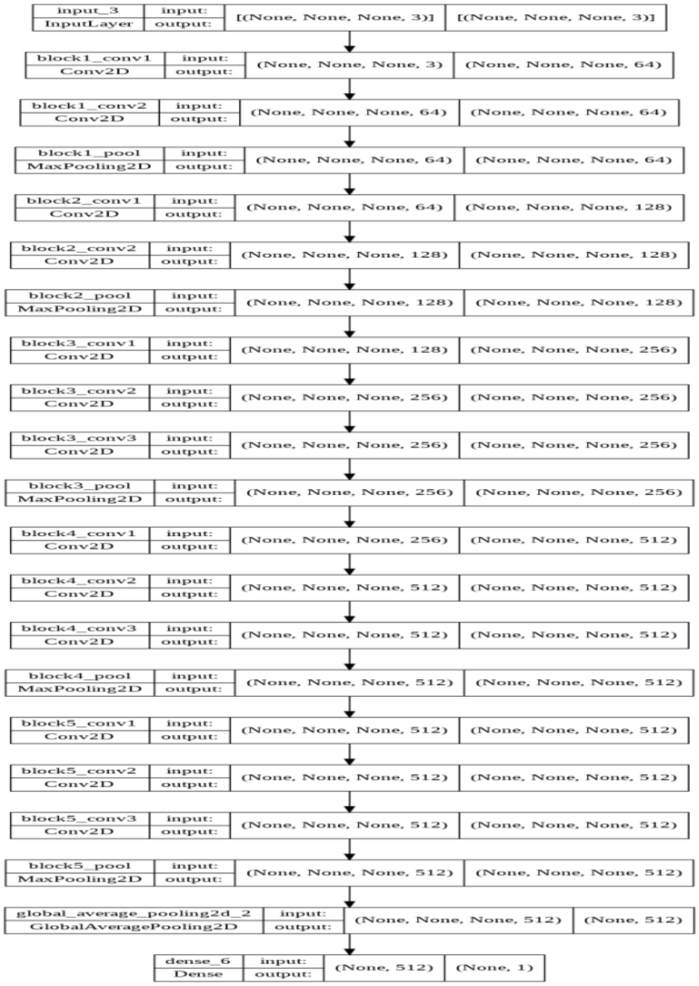
VGG16 for Image

This module deals with detection of offensive memes using Image only. First, we do some preprocessing on the image. Some preprocessing techniques used include Pixel brightness transformations/ Brightness corrections, Geometric Transformations, Image Filtering and Segmentation, Fourier transform and Image restauration.

Then this processed image is fed as input to the pre-trained CNN model. The architecture used in the CNN model is VGG-16. The model has been pre-trained on the IMAGENET dataset. All layers' weights of the model except the last 2 layers have been frozen. A Global Average Pooling Layer and a Dense Layer has been added at the end.

The last 2 layers of the model have been trained on the training dataset. The prediction is done on the test dataset. Finally we evaluate our trained model using techniques like confusion matrix and also with metrics like precision, recall, f-score and accuracy.

VGG-16 Architecture

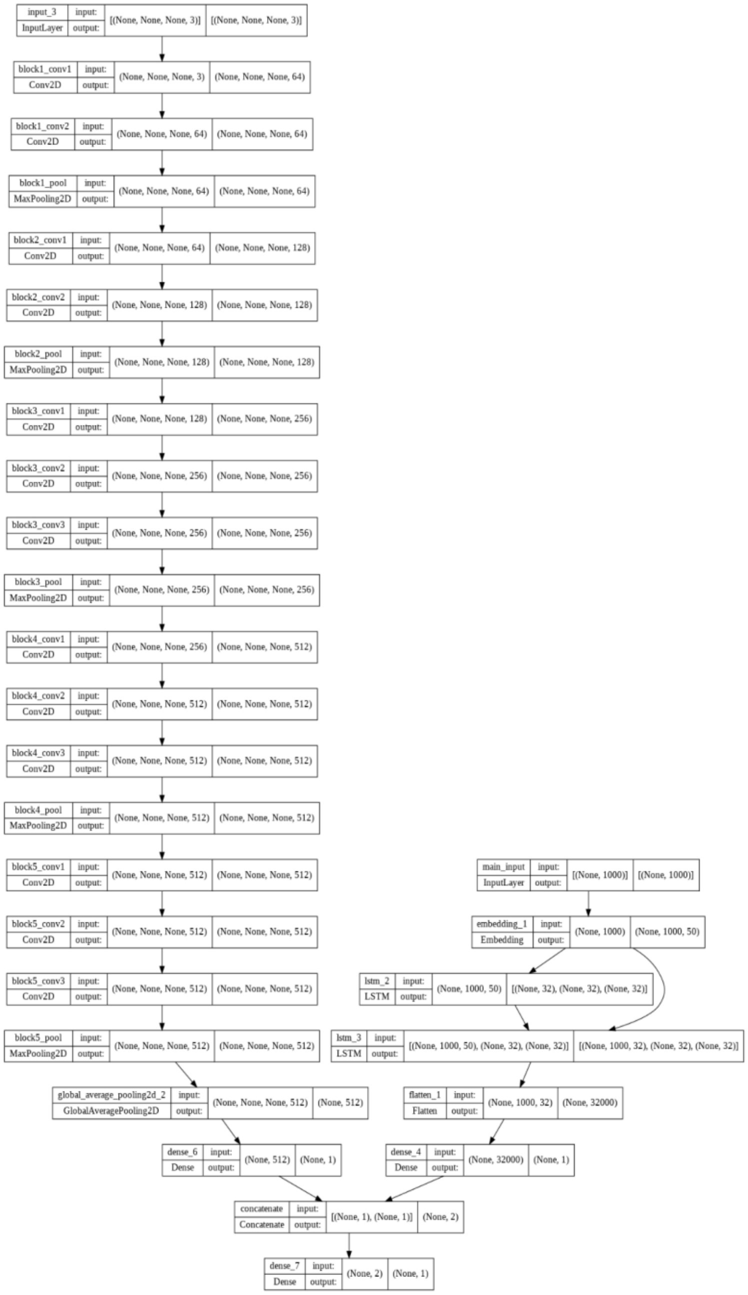

input_3 InputLayer input: output: [(None, None, None, 3)] [(None, None, None, 3)]
block1_conv1 Conv2D input: output: (None, None, None, 3) (None, None, None, 64)
block1_conv2 Conv2D input: output: (None, None, None, 64) (None, None, None, 64)
block1_pool MaxPooling2D input: output: (None, None, None, 64) (None, None, None, 64)
block2_conv1 Conv2D input: output: (None, None, None, 64) (None, None, None, 128)
block2_conv2 Conv2D input: output: (None, None, None, 128) (None, None, None, 128)
block2_pool MaxPooling2D input: output: (None, None, None, 128) (None, None, None, 128)
block3_conv1 Conv2D input: output: (None, None, None, 128) (None, None, None, 256)
block3_conv2 Conv2D input: output: (None, None, None, 256) (None, None, None, 256)
block3_conv3 Conv2D input: output: (None, None, None, 256) (None, None, None, 256)
block3_pool MaxPooling2D input: output: (None, None, None, 256) (None, None, None, 256)
block4_conv1 Conv2D input: output: (None, None, None, 256) (None, None, None, 512)
block4_conv2 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block4_conv3 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block4_pool MaxPooling2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_conv1 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_conv2 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_conv3 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_pool MaxPooling2D input: output: (None, None, None, 512) (None, None, None, 512)
global_average_pooling2d_2 GlobalAveragePooling2D input: output: (None, None, None, 512) (None, 512)
dense_6 Dense input: output: (None, 512) (None, 1)

Combined Model

After Working on image data and textual data individually , we have used three previously used classifiers combined with VGG16 to predict the output class for the meme as a whole . Pre -trained VGG16 on the ImageNet dataset is used with the GloVe algorithm to represent word embeddings. The three combined models proposed are :

Stacked LSTM +VGG16

Stacked LSTM which is a LSTM model composed of several LSTM layers is also used to work on textual data , here we have used the same stacked LSTM model for the combined approach along with pre-trained VGG16.


input_3 InputLayer input: output: [(None, None, None, 3)] [(None, None, None, 3)]
block1_conv1 Conv2D input: output: (None, None, None, 3) (None, None, None, 64)
block1_conv2 Conv2D input: output: (None, None, None, 64) (None, None, None, 64)
block1_pool MaxPooling2D input: output: (None, None, None, 64) (None, None, None, 64)
block2_conv1 Conv2D input: output: (None, None, None, 64) (None, None, None, 128)
block2_conv2 Conv2D input: output: (None, None, None, 128) (None, None, None, 128)
block2_pool MaxPooling2D input: output: (None, None, None, 128) (None, None, None, 128)
block3_conv1 Conv2D input: output: (None, None, None, 128) (None, None, None, 256)
block3_conv2 Conv2D input: output: (None, None, None, 256) (None, None, None, 256)
block3_conv3 Conv2D input: output: (None, None, None, 256) (None, None, None, 256)
block3_pool MaxPooling2D input: output: (None, None, None, 256) (None, None, None, 256)
block4_conv1 Conv2D input: output: (None, None, None, 256) (None, None, None, 512)
block4_conv2 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block4_conv3 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block4_pool MaxPooling2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_conv1 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_conv2 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_conv3 Conv2D input: output: (None, None, None, 512) (None, None, None, 512)
block5_pool MaxPooling2D input: output: (None, None, None, 512) (None, None, None, 512)
main_input InputLayer input: output: [(None, 1000)] [(None, 1000)]
embedding_1 Embedding input: output: (None, 1000) (None, 1000, 50)
lstm_2 LSTM input: output: (None, 1000, 50) [(None, 32), (None, 32), (None, 32)]
lstm_3 LSTM input: output: [(None, 1000, 50), (None, 32), (None, 32)] [(None, 1000, 32), (None, 32), (None, 32)]
global_average_pooling2d_2 GlobalAveragePooling2D input: output: (None, None, None, 512) (None, 512)
flatten_1 Flatten input: output: (None, 1000, 32) (None, 32000)
dense_6 Dense input: output: (None, 512) (None, 1)
dense_4 Dense input: output: (None, 32000) (None, 1)
concatenate Concatenate input: output: [(None, 1), (None, 1)] (None, 2)
dense_7 Dense input: output: (None, 2) (None, 1)

7.References

[Base paper] Balakrishnan, V., Ng, K. S., & Arabnia, H. R. (2022). Unravelling social media racial discriminations through a semi-supervised approach. *Telematics and Informatics*, *67*, 101752. [link]

[1] S. Khan et al., "HCovBi-Caps: Hate Speech Detection Using Convolutional and Bi-Directional Gated Recurrent Unit With Capsule Network," in IEEE Access, vol. 10, pp. 7881-7894, 2022, doi: 10.1109/ACCESS.2022.3143799. [link]

[2] E. Lee, F. Rustam, P. B. Washington, F. E. Barakaz, W. Aljedaani and I. Ashraf, "Racism Detection by Analysing Differential Opinions Through Sentiment Analysis of Tweets Using Stacked Ensemble GCR-NN Model," in IEEE Access, vol. 10, pp. 9717-9728, 2022, doi: 10.1109/ACCESS.2022.3144266. [Link]

[3] S. Ni, J. Li and H. -Y. Kao, "MVAN: Multi-View Attention Networks for Fake News Detection on Social Media," in IEEE Access, vol. 9, pp. 106907-106917, 2021, doi: 10.1109/ACCESS.2021.3100245. [Link]

[4] A. T. Aind, A. Ramnaney and D. Sethia, "Q-Bully: A Reinforcement Learning based Cyberbullying Detection Framework," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154092. [Link]

[5] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017, April). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion* (pp. 759-760). [Link]

[6] Öztürk, N., & Ayvaz, S. (2018). Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis. *Telematics and Informatics*, *35*(1), 136-147. [Link]

[7] Ozbay, F. A., & Alatas, B. (2020). Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and its Applications*, *540*, 123174. [Link]

[8] Murnion, S., Buchanan, W. J., Smales, A., & Russell, G. (2018). Machine learning and semantic analysis of in-game chat for cyberbullying. *Computers & Security*, *76*, 197-213. [link]

[9] Rizoiu, Marian-Andrei & Wang, Tianyu & Ferraro, Gabriela & Suominen, Hanna. (2019). Transfer Learning for Hate Speech Detection in Social Media. [link]

[10] P. K. Roy, A. K. Tripathy, T. K. Das and X. -Z. Gao, "A Framework for Hate Speech Detection Using Deep Convolutional Neural Network," in IEEE Access, vol. 8, pp. 204951-204962, 2020, doi: 10.1109/ACCESS.2020.3037073. [link]

[11] Agarwal, S., & Chowdary, C. R. (2021). Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. Expert Systems with Applications, 185, 115632. doi:10.1016/j.eswa.2021.115632 [link]

[12] Kumar, A., & Sachdeva, N. (2021). Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network. Multimedia Systems. doi:10.1007/s00530-020-00747-5 [link]

8.Video Link -

https://drive.google.com/drive/folders/1wUKtpjgYZsU3aKOpvXeg_Z6-zBQHkpZI?usp=sharing

--THE END--