HATE SPEECH DETECTION USING MTL

by

B.MAHATHI 421122

B.SOWJANYA 421117

D.S.HARSHITHA 421137

Under the guidance of

Dr. K. HIMABINDU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534101, INDIA

MAY 2024

This is page is left blank

HATE SPEECH DETECTION USING MTL

Thesis submitted to
National Institute of Technology Andhra Pradesh
for the award of the degree

of

Bachelor of Technology

by

B.MAHATHI 421122

B.SOWJANYA 421117

D.S.HARSHITHA 421137

Under the guidance of

DR.K.HIMABINDU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

TADEPALLIGUDEM-534101, INDIA

MAY 2024

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

B.MAHATHI	B.SOWJANYA	D.S.HARSHITHA
421122	421117	421137
Date:_____	Date:_____	Date:_____

CERTIFICATE

It is certified that the work contained in the thesis titled **HATE SPEECH DETECTION USING MTL** by B.MAHATHI, bearing Roll No: 421122, B.SOWJANYA, bearing Roll No: 421117 and D.S.HARSHITHA, bearing Roll No: 421137 has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature
Dr.K.HIMABINDU
Department of CSE
N.I.T. Andhra Pradesh
MAY, 2024

ACKNOWLEDGEMENT

We want to express our sincere gratitude to our supervisor Dr.K.HimaBindu ma'am, who have showed us the path on how to proceed through this project and made us to overcome our weaknesses and difficulties while completing this project.we also want to thank Mr.Kiran sir for helping us whenever we got stuck through the code without which this task would have never been accomplished.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this decade, the number of people using social media has skyrocketed, encompassing all age, ethnic, color, and gender groups. Now a days these platforms have become hubs filled with hatred and offensive language. As, manually filtering data from social media where around millions of posts are posted every second is impossible, a more automatic approach is needed. Many models have been developed for hate speech detection on social media posts. Though most of them are based on Single-Task learning approach where optimization of weights of a model to perform well in a single task, our project explores the use of MTL (Multi-Task Learning) approach where there are multiple objectives which are need to be optimized to identify hate speech in Twitter tweets. Investigation about the relationships between hate speech and similar concepts 1) Hate speech with polarity 2) Hate speech with emotion 3) Hate speech with polarity and emotion has done. So, four different models were trained: STL (Single-Task Learning), MTL with emotion, MTL with polarity, and MTL with both emotion and polarity on hateEval, Emoevent, twitter sentiment analysis datasets. Our results demonstrated that incorporating both polarity and emotion knowledge enhances hate speech detection accuracy compared to the STL approach. Based on the experimentation on the datasets used, MTL models built by us, proved their efficiency by increasing the F1 score by 4% when compared to the STL model built by us.

TABLE OF CONTENTS

Contents

1 Introduction

In this decade the number of persons using social media have sky rocketed. and the users include all age, ethnic, color, gender groups.With the rise in social media it's been very easy to share one's thoughts and perspectives and its reaching millions of people in just some seconds. Though it looks like a powerful tool to build strong communication among humans it also servers as great threat if its being used in bad way. Now a days these social media platforms have become a hub for hatred, offensive language. The European Commission's recommendation against racism and intolerance has a definition for HOF as "the advocacy, promotion or incitement of the denigration, hatred or vilification of a person or group of persons, as well any harassment, insult, negative stereotyping, stigmatization or threat of such person or persons and any justification of all these forms of expression– that is based on a non-exhaustive list of personal characteristics or status that includes 'race', color, language, religion or belief, nationality or national or ethnic origin, as well as descent, age, disability, sex, gender, gender identity, and sexual orientation.

Today, the vast and uncontrolled content posted every day on the Web makes difficult and impractical to track the content of comments manually. In order to prevent the spread of hate speech online, in May 2016, the Eu ropean Commission made an agreement with Facebook, Microsoft, Twitter and YouTube a "Code of conduct on countering illegal HS online"2. During 2018, Instagram, Snapchat and Dailymotion joined the Code of Conduct. Jeuxvideo.com joined in January 2019, and TikTok announced their participation to the Code in September 2020. But, it is not easy to fulfill. Here NLP researchers came to their rescue. They developed many models in order to detect hate speech but most of them are single task optimization (STL) model where model is trained for single task only. Our base paper authors have come with a Multi Task Learning (MTL) approach where one model is trained to solve more than one task, here model has more than one objective functions to optimize.

To check weather MTL helps in detecting hate speech, 4 model were developed to investigate the relation between various factors. 1) STL approach 2) MTL with emotion which explores the relation between hate speech and emotion detection 3) MTL with polarity which explores the relation between hate speech and polarity 4) MTL with both emotion and polarity which explores the relation among hate speech, emotion and polarity. Polarity classification focuses on high level overview of a statement weather it is positive or negative sentiment. Where as, sentiment classification is more fine grained task where model needs to predict a emotion in anger, fear, sadness, joy, surprise, and disgust. Its not suggested to mix random tasks in MTL approach, as negative sentiment is often indicator of emotions like anger, sadness, fear etc. and hate speech is highly correlation with the negativity of the statement, authors have planned to build a model which optimizes all 3 tasks. Based on our results it's observed that the knowledge attained by through other tasks i.e emotion, polarity helped the model to detect hate speech with more accuracy. Models with MTL approach have surpassed model with STL approach.

2 Literature Review

2.1 Single task learning approach

While NLP researches have developed many models to detect hate speech. The initial phase of research was majorly done in machine learning algorithms like Support Vector Machines, Random Forest, Decision Tree and Logistic Regression along with the combination of different types of syntactic, semantic, lexical, sentiment, and lexicon-based features. Here, suitable features for hate speech detection and needed to be found and given to the model. Though they were encountering some good results there was lot of scope to improve.

After some years interest of research was shifted to neural networks based architecture like CNN[7], RNN [8]etc. models like For example, in order to break the barrier of language dependency in word embedding approach, researchers conducted an ensemble of RNN classifiers[2], incorporating various features associated with user related information. Some authors experimented with a robust system based on compositional RNNs able to handle even substantially noisy inputs, and reached competitive results for HS detection in English texts. Then in transformers era many models like BERT based, hateBERT, RoBERTa showed very promising results.

2.2 Emotion analysis in hate speech

Emotion analysis offers a valuable tool that helps to enhance the performance of machine learning classification systems. A few recent studies have investigated the benefit of using emotion features for HS detection. For instance, some authors follow the idea that the concept of HS can be split into two main components hate and speech, and based on this they proposed a new definition of HS in the scope of emotional analysis: "any emotional expression imparting opinions or ideas bringing a subjective opinion or idea to an external audience with discriminatory purposes". As, hate speech is highly correlated with emotions like anger, disgust many authors like authors have increased accuracy in hate speech detection leveraging the knowledge gained from sentiment analysis task. Authors in have [2] also explored the effect of emotion in hate speech and offensive language they also suggested that their model beat mBert and Bert for hate speech detection.

2.3 Polarity analysis in hate speech

Polarity detection has emerged as one of the most well-known areas in NLP due to its significant implications in social media mining. A negative sentiment can be an indicator of the presence of offensive language. Sentiment analysis and the identification of HOF share common discursive properties. Considering the example in [2] , "I am sick and tired of this stupid situation", in addition to expressing anger, conveys a negative sentiment along with the presence of expletive language targeted to a situation. Therefore, both sentiment and emotion features can be used as useful information in the NLP systems to benefit the task of HOF detection in social media. Unlike EA, as SA classification is one of the most studied tasks due to its broader applications, a larger number of corpora annotated with sentiments is available, particularly from Twitter. For instance, one of the most well-known datasets is the Stanford Sentiment Treebank.

2.4 Multi task learning approach

Though our base model uses MTL approach they used different datasets for different tasks which is not the ideal case, authors[2] explained how to handle different datasets and simulate MTL using those datasets. The following method has followed in their research paper, a mini-batch Bt is selected among all 4 tasks, and the model is updated according to the task-specific objective for the task t. This approximately optimizes the sum of all multi-task objectives. Many authors like [3][4][5] inspired others and stated that MTL approach can be used as a valid and useful approach to increase the accuracy of targeted task by using the knowledge attained by other in multi task approach provided all tasks that are taken are well related to each other.

3 Datasets

Here, models were built based to solve 3 different tasks namely hate speech detection, emotion classification and polarity classification. So, different datasets were used for each task.

3.1 Hate speech

For training model to detect hate speech in twitter dataset which is our main objective, HateEval 2018 task A datasets were used as our train, development and test datasets. Task A in HateEval 2018 is : Hate Speech Detection against Immigrants and Women: a two-class (or binary) classification where systems have to predict whether a tweet in English or in Spanish with a given target (women or immigrants) is hateful or not hateful. The column headings of this dataset are 'id' : represents the no of the tweet, 'text' : represents tweet, 'HS', : 1 for hate speech, 0 for not a hate speech, 'TR' : representing target women or immigrants, 'AG' : 1 for aggressive and 0 for not an aggressive tweet. Out of 9100 data samples, 3833 of them are tagged as hate speech and the remaining 5267 are tagged as not a hate speech.

3.2 Emotion classification

For training model to classify emotion in twitter dataset, EmoEvent datasets were used as our train development and test datasets. EmoEvent is a multilingual emotion dataset based on events that took place in April 2019. It focuses on tweets in the areas of entertainment, catastrophes, politics, global commemoration and global strikes. The authors collected Spanish and English tweets from the Twitter platform. Then, each tweet was labeled with one of seven emotions, six Ekman's basic emotions plus the "neutral or other emotions" label.Here other emotions include "joy", "sadness", "anger", "suprise", "disgust", and "fear". The labeling was done by three Amazon Mechanical Turkers. The column headings of this dataset are "id" : to uniquely indentify each tweet, "tweet": represents the tweet, "emotion": one in ["joy", "sadness", "anger", "suprise", "disgust", "fear", "others"], "offensive" : 1 if offensive 0 if not.

3.3 Polarity classification

For training model to classify polarity in twitter dataset, Twitter sentiment analysis dataset in kaggle was used as our train, test, development datasets. The column headings of the dataset includes "id" : a unique no to identify tweet, "entity" , "sentiment" : "postive" for tweets with positive sentences "negative" for tweets with negative sentences, "tweet content" : whole tweet. There are 5647 positive, negative and neutral sentences in this dataset.

4 Proposed Methodology

4.1 Model

STL

STL is a paradigm that updates the weight of neural networks using the input sequence of a single classification task involving a dataset. In order to establish a baseline in our study and compare the results with MTL scenario, STL model that use the HS task as the sole optimization objective was taken. In single task learning approach, the input is a tweet first it is passed through BERT to get vector representation of words which are contextual aware. Then these vectors are passed to a neural network architecture which gives output 1 for hate speech, 0 for not a hate speech.

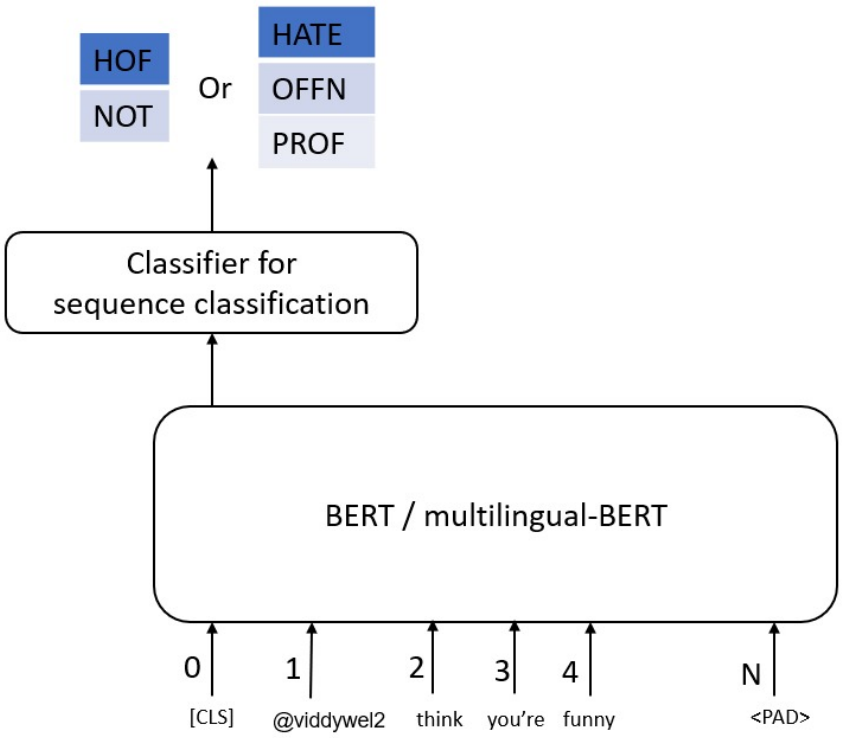

Figure 1. STL approch

MTL

To understand MTL, a good understanding of transfer learning is needed first. Transfer learning at its core means using knowledge of one task to leverage another task.There are 2 kinds of transfer learning paradigm. Those are 1) Sequential transfer learning: where one model is trained first on a pariticular task, then the same model is used for another task. 2) Parllel transfer learning: where knowledge is transferred simultaneously from multiple source tasks to a single target task. In this paradigm, instead of transferring knowledge sequentially from one task to another, information from multiple source tasks is utilized concurrently to improve the learning performance of the target task. Our proposed MTL approach follows parallel transfer approach.

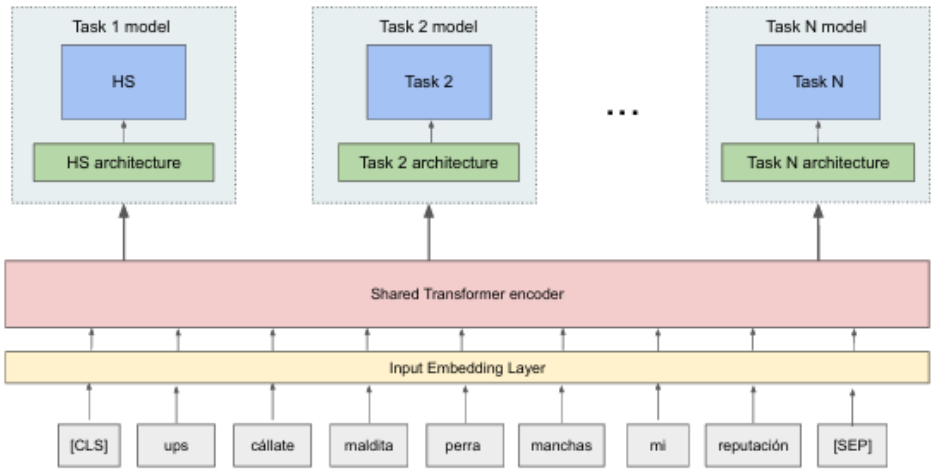

Figure 2. MTL approch [6]

Parameter sharing is a core concept of MTL approach. There are two types of parameter sharing that can happen. They are 1) Soft parameter sharing : In this, for every task a separate model was built with no parameters being shared directly. The distance between the parameters of the model is regularized in order to encourage similarity between the parameters. 2) Hard parameter sharing : This technique is the most widely used MTL approach in neural networks. It consists of a single encoder that is shared and updated between all tasks, while keeping several specific heads for each task.

Fig 2 shows our model architecture i.e is MTL with emotion and polarity. There is only one input layer which takes input tweet. Then comes the shared encoder. Here the shared encoder is Bidirectional Encoder Representations from Transformers [BERT] which gives contextual embedding vectors to the input sentence. Then, 3 task specific heads each for a task namely polarity classification, sentiment classification and hate speech detection were built on top of it.

As, said earlier 4 different models were developed.3 out of them belongs to MTL approach. MTL with polarity, MTL with emotion has the same architecture as of fig 1 but they have only 2 task specific heads.

4.2 Methodology

Data preprocessing

The input tweet is being feed into BERT model. As these are tweets it contains lot's of noise like hashtags. emojies, user names etc. Twitter related data cleaning has done. Tweets have several problems because they are written in a regional language and accents. Therefore, there are numerous challenges in tokenizing tweets such as duplication, usage of informal terms, noise ,user mentions,hashtags and emojis.So, all these problems are needed to be addressed in data pre processing. regex was used to do all prepocessing. For this NLTK, emoji libraries to implement this part.

- replacing all urls with the word "url"
- replacing all usernames with the word "user"
- replacing all emails with the word "email"
- replacing all monitory, dates, phone no, time with their names respectively
- replacing all emojies with their corresponding names using emoji library
- remove all stop words using stop words library

Shared encoder - BERT

The processed data is then feed into BERT. First BERT tokeniser that is word piece tokeniser tokenises input sentences. Then, Bert adds 2 extra tokens namely [CLS], [SEP] for every input sentence. Fig 3 shows the architecture of BERT. The outputs of BERT are last hidden states, polled outputs, last hidden states[0][0] contains CLS token embeddings, which is what is being feed into task specific heads. CLS token of BERT captures semantic meaning of entire sentence, which is why for classification tasks it is mostly used.

$$In = \{[CLS], w1, w2, w3...[SEP]\}$$

$$0u = \{H_{\text{CLS}}, H1, H2, H3..H_{\text{SEP}}\}$$

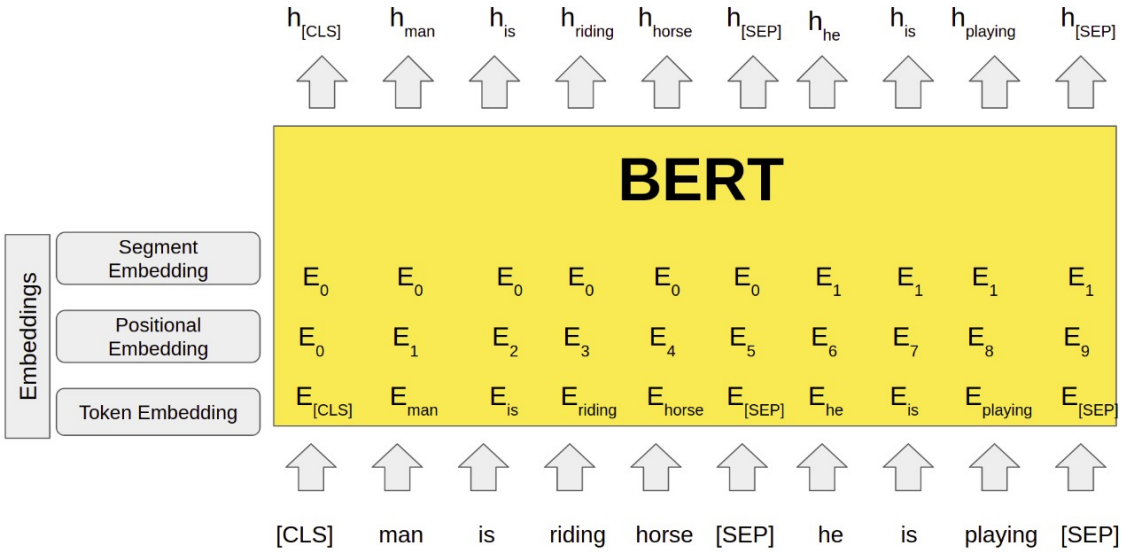

Figure 3. BERT

BERT has a transformer architecture, trained on masked language modelling and next sentence prediction. Masked language modelling means random masking some of the tokens and the transformers should output correct set of tokens. 15% of input tokens are masked. Out of them 80% are replaced by [MASK] token, 10% of them are replaced by new tokens, and remaining 10% are stayed as it is. Next sentence prediction is a binary classification task where a model will be given with 2 sentences and model should output 1 if one sentence follows other, else 0.

Task specific heads

The context aware vectors embeddings from BERT are then passed to task specific heads as shown in fig 2. In ideal case where every sentence has all tasks labels, so context aware embeddings will be sent to every task specific head. But, that's not in our case. Different datasets have been used for different tasks, so embeddings obtained from BERT can't be passed to all task specific code. It will be sent to the corresponding task specific head only, this means, an input sample of polarity detection then the embeddings will only be sent to the polarity detection head. in cases like this simulating MTL approach is not as straight forward. The following method has adopted to tackle this problem. In the MTL stage, during each epoch, a mini-batch b_t is selected among all 4 tasks, and the model is updated according to the task-specific objective for the task t. This approximately optimizes the sum of all multi-task objectives. The output of each task specific head is then passed to a softmax layer, the predicted label is the one the which is having highest probability

Loss will be calculated for every task specific head. Total loss is summation of all individual task loses, and this total loss is back propagated.In our model, all the parameters in the shared encoder will be updated based on every task loss where as task specific heads parameters are only updated based on their respective task loss.

$$L = \sum_{i=1}^{n} l_i$$

$$y^t = Softmax(W^t.H_{\text{CLS}} + b_t)$$

$$y_t = \arg\max y^t$$

Algorithm

Algorithm 1 MTL with emotion and polarity Algorithm

Initialize the learning parameters and hyper parameters
Initialize the optimizer and loss function
for $epoch = 1$ to $EPOCHS$ **do**
  Initialize task no as 0
  $Loss < -0$
  for $batch = 1$ to $BATCHES$ **do**
    Get batch from the Polarity dataloader B_p
    Initialize task no as 1
    $Ou < -BERT(B_p, taskno)$
    $y^1 < -Softmax(W^t.H_{\text{CLS}} + b_1)$
    $l_1 <$ −Calculate loss between predicted probabilities and gold labels
    $Loss = Loss + l_1$
    Get batch from the Emotion dataloader B_e
    Initialize task no as 2
    $Ou < -BERT(B_e, taskno)$
    $y^2 < -Softmax(W^t.H_{\text{CLS}} + b_2)$
    $l_2 <$ −Calculate loss between predicted probabilities and gold labels
    $Loss = Loss + l_2$
    Get batch from the Hate sppeech dataloader B_h
    Initialize task no as 3
    $Ou < -BERT(B_h, taskno)$
    $y^3 < -Softmax(W^t.H_{\text{CLS}} + b_3)$
    $l_3 <$ −Calculate loss between predicted probabilities and gold labels
    $Loss = Loss + l_3$
    Update model parameters using the optimizer using back propagation algorithm in order to minimise loss
  end for
  Compute average loss for the epoch
  Evaluate the model on the validation set
end for

5 Experimental procedure

All the models were implemented using PyTorch, a high performance deep learning library based on the Torch library. Training of models was done on train set and then tested it HateEval test set. The experiments were run on a P100 GPU on kaggle, we couldn't run it on Google collab as the computational and memory resources were not enough.

5.1 Hyper paramters

Batch size has taken as 8, number of epochs as 3, the bert layers were freezed for first 2 epochs, then unfreezing it in the last epoch. Cross entropy loss was used as our loss function for all 3 task specific heads . AdamW optimizer with learning rate of 1e-4 and weight decay of 0.01 was used as the optimizer for every task specific head. Torch eval library in pytorch library was used to calculate evaluation metrics.

Model	Parameters
STL	10,96,76,642
MTL with polarity	10,98,71,245
MTL with emotion	10,98,72,049
MTL with polarity and emotion	11,00,66,652

Table 1: No of learnable parameters

5.2 Results

F1 scores for all 4 models were computed. As, f1 is more useful metric accuracy, precision, and recall. Our results are as following:

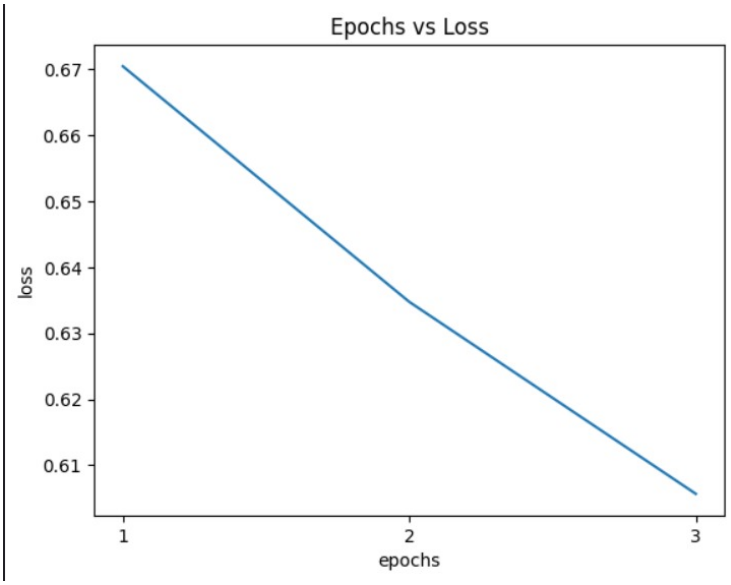

Figure 4. STL loss curve

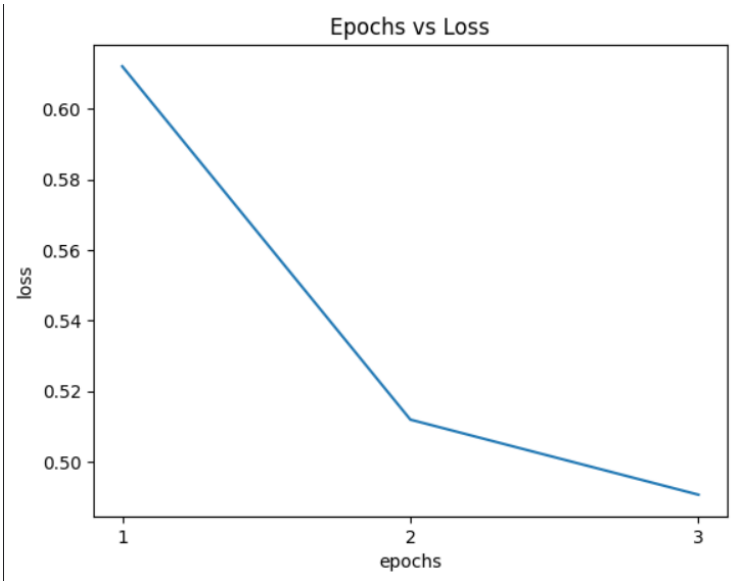

Figure 5. MTL with polarity and emotion loss curve

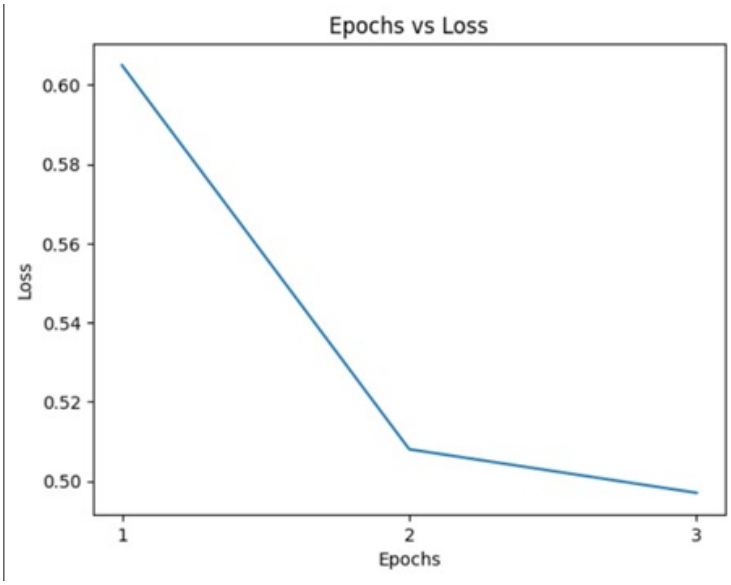

Figure 6. MTL with polarity loss curve

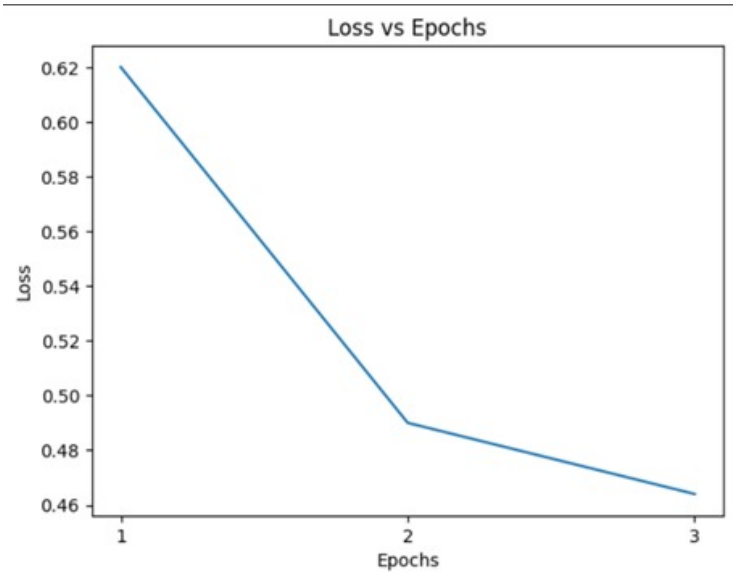

Figure 7. MTL with emotion loss curve

Approach	**F1**
CNN-FastText[7]	48.8%
RNN based UTFPR/W [8]	50.9%
RNN based UTFPR/O [8]	57.0%
Bi-LSTM [9]	46.6%
Our STL model	66.4%

Table 2: STL models comparision

Based on this results it can be infered that, knowledge obtained from tasks like polarity classification and emotion classification has helped in getting more accuracy than STL model.

	Our results
MTL with emotion	71.47
MTL with polarity	70.50
MTL with polarity and emotion	72.27

Table 3: MTL results Table

5.3 Error analysis

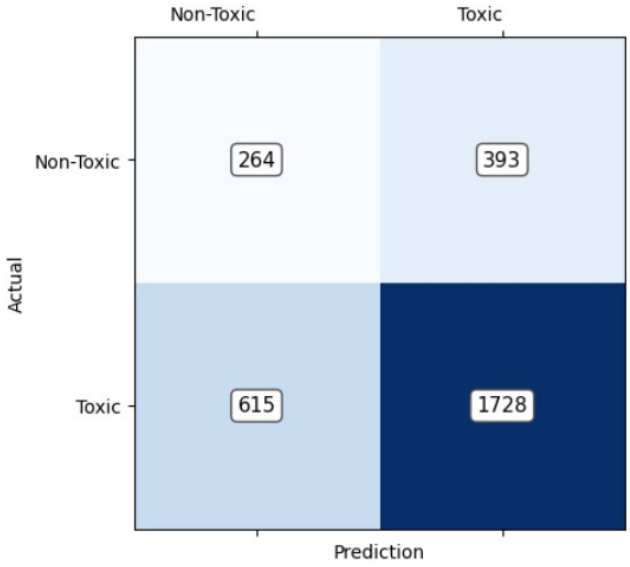

Figure 10. STL confusion matrix

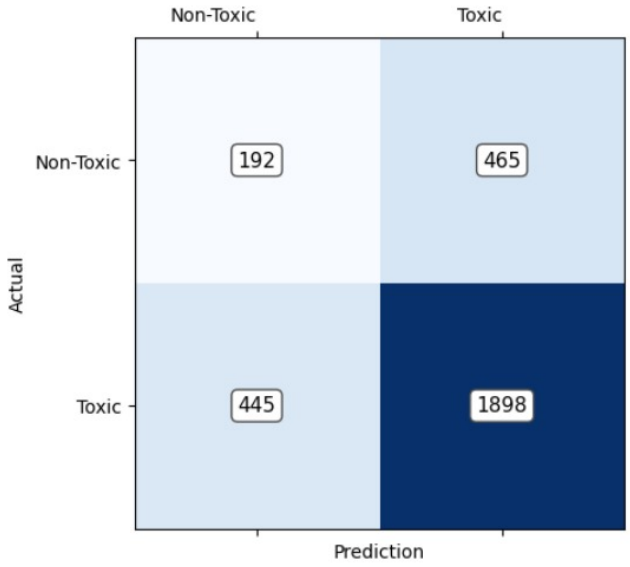

Figure 11. MTL with emotion and polarity confusion matrix

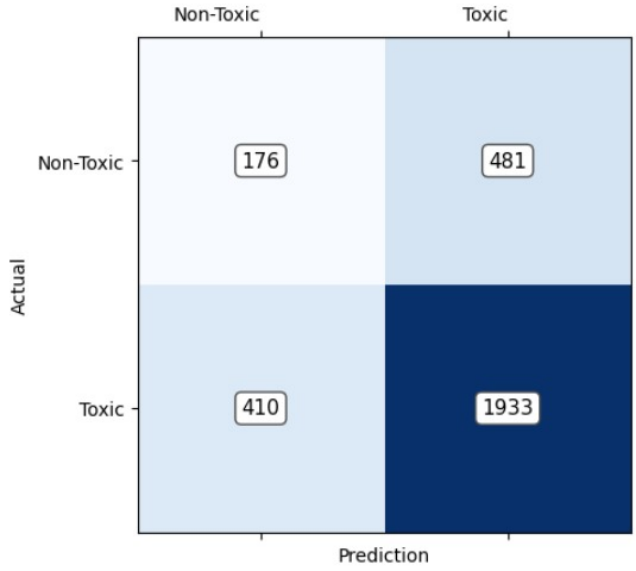

Figure 12. MTL with polarity confusion matrix

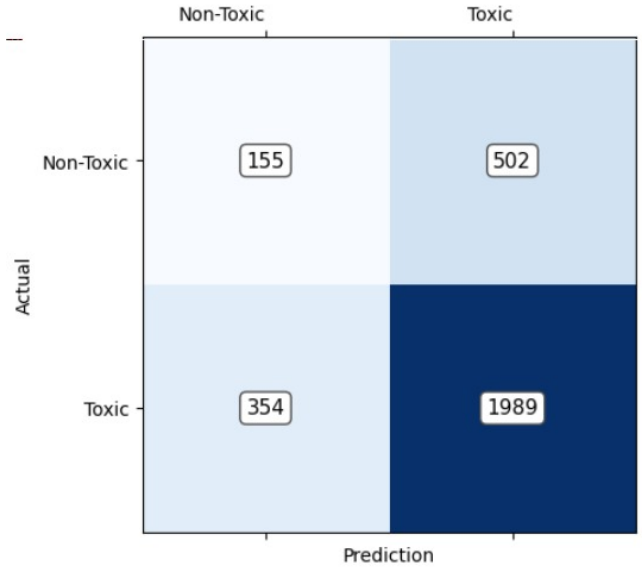

Figure 13. MTL with emotion confusion matrix

sentence	**gold label**	**STL**	**MTL with emotion and polarity**
unburn bitch whoever invents spf 1000000 willlove forever cuz need shit url	0	0	1
gonna lock asks ? bitch i'll eat pizza tomorrow cause friday hoe	0	0	1
got whole eyeshadow palette three eyebrow pencils ready bad bitch shsyxgxjx	0	1	0
love romantic date mallorca , reminds lauren's date palma city silly bitch i'm still jealous . even cunt	0	1	0

Table 4: STL vs MTL with polarity and emotion

To analyse errors, collection of false positives(not hate speech predicted as hate speech) and false negatives(hate speech but predicted as not a hate speech) for STL and MTL with emotion and polarity models was done. A close

look on both of the false positives revealed that the number of times negative words like ”bitch”, ”hoe”, ”ass” appearing in MTL’s false positives (bitch - 250, hoe - 71, ass - 52) is more than their appearance in STL’s false positives (bitch - 203, hoe - 43, ass - 36). As, the presence of these words makes the model to classify sentences as negative polarity and anger this lead to hate speech as prediction. On the other hand in table 4, for the last 2 sentences MTL model predicted it as not a hate speech because those sentences were classified as neutral emotion, neutral and negative polarity respectively.

5.4 GUI

Demonstration of our model was done by building a GUI for it where user can give input sentences in english in the left side and output will be shown on the right side. This GUI was developed using Gradio library. Gradio is a python library which enables us to build a customised GUI within minutes for ML, DL and data applications. It provides an easy to use interface for the user to interact with deep learning models. Below are the screen shots of the outputs from the website.

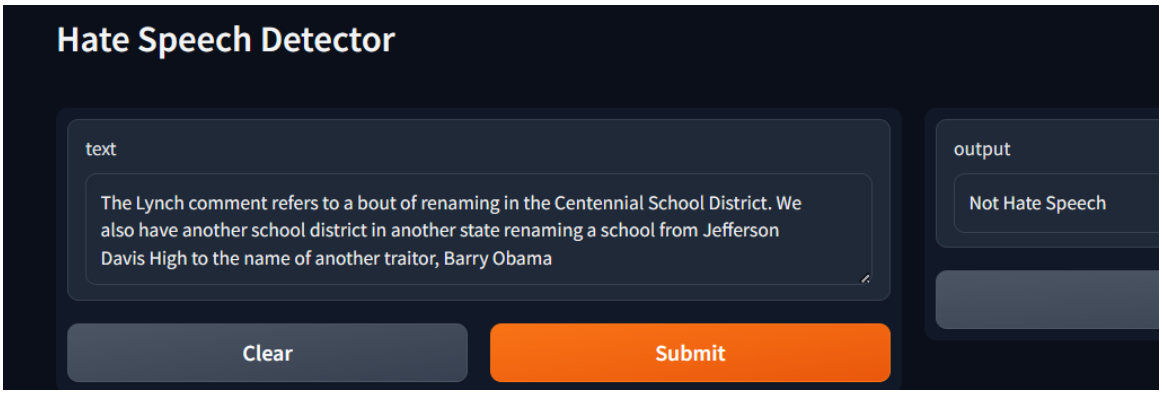

Figure 8. Figure showing output as not a hate speech

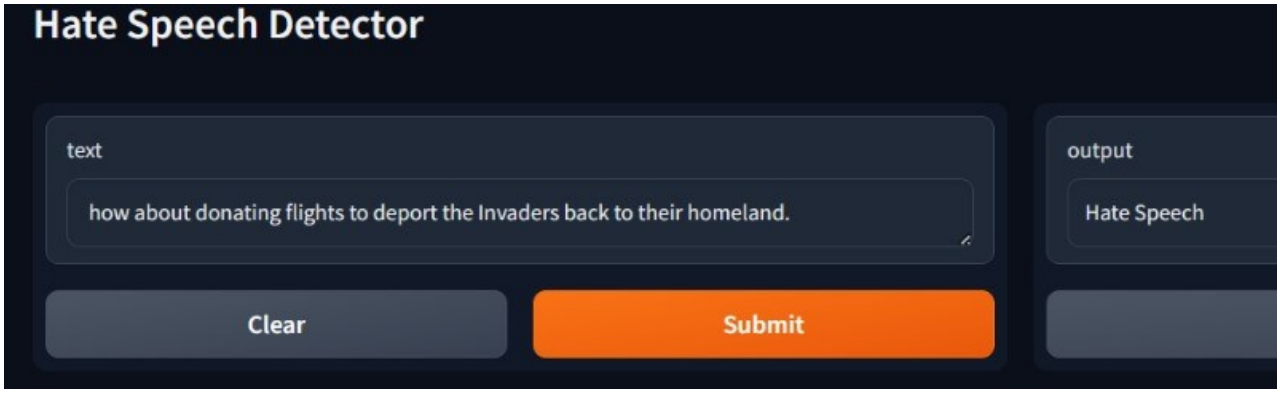

Figure 9. Figure showing output as hate speech

6 Conclusion and Future work

As, the amount of hate speech is increasing in the order of millions no of post every day an attempt to automate the filtering of hate speech in social media was done .Our attempt to use MTL approach to find hate speech in twitter posts was successful, as it have much higher accuracy than than model with single optimisation task. There are many advantages of using MTL approach such as it helps to improve generalisation as the model need to a suitable

representation which works for all the tasks, it handles data inefficiency problem even though having less amount of data for one task because as training with all tasks data was eone, moreover it does regularization and addresses label imbalance. In practise not all combination of tasks works that why all 3 models were build in MTL paradigm, there can be negative transfer in the place of positive transfer which leads to decrease in the accuracy. So, choosing the correct combination of tasks is really important.

As, future work we want to explore how irony, sarcasm also effects hate speech detection. Because due to sarcasm model can understand the over all sentiment as positive because understand sarcasm really needs a greater knowledge about the language and same goes with the irony also. In this 2 cases, even though the over all sentiment outputted by model is positive we need label as hate speech. Also, we want to explore about how can we deal with code mixed data, and for some Telugu regional languages with MLT.

7 References

[1]Multi-Task Learning with Sentiment, Emotion, and Target Detection to Recognize Hate Speech and Offensive Language Flor Miriam Plaza-del-Arco1,3, Sercan Halat2,3, Sebastian Padó3 and Roman Klinger3 https://ceur-ws.org/Vol-3159/T1-30.pdf

[2]Hate Speech and Offensive Language Detection using an Emotion-aware Shared Encoder Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, Noel Crespi https://arxiv.org/abs/2302.08777

[3]Multi-task Learning for Hate Speech and Aggression Detection Faneva RAMIANDRISOA1 https://ceur-ws.org/Vol-3178/CIRCLE_2022_paper_31.pdf

[4]T5 for Hate Speech, Augmented Data and Ensemble Tosin Adewumi, Sana Sabah Sabry, Nosheen Abid, Foteini Liwicki Marcus Liwicki MLGroup, EISLAB, Luleå University of Technology, Swedenhttps://arxiv.org/abs/2210.05480

[5]An overview of multi-task learningYu Zhangand Qiang Yanghttps://www.researchgate.net/publication/323599831_An_overview_of_multi-task_learning

[6]A Multi-Task Learning Approach to Hate Speech Detection Leveraging Sentiment Analysishttps://ieeexplore
ieee.org/document/9509436 [7]NF-HatEval at SemEval-2019 Task 5: Convolutional Neural Networks for Hate Speech Detection Against Women and Immigrants on Twitterhttps://aclanthology.org/S19-2074/

[8]UTFPRatSemEval-2019 Task 5: Hate Speech Identification with Recurrent Neural Networkhttps://arxiv.org/abs/1904.07839

[9]ABARUAH at SemEval-2019 Task 5 : Bi-directional LSTM for Hate Speech Detection Arup Baruah, Ferdous Barbhuiya, Kuntal Deyhttps://aclanthology.org/S19-2065/